This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2025.3531349

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2023.0322000

Enhancing Democratic Processes: A Survey of DRE, Internet, and Blockchain in Electronic Voting Systems

MOSBAH ALOWN[1], MEHMET SABIR KIRAZ[1], and MUHAMMED ALI BINGOL[1]
[1]Cyber Technology Institute, De Montfort University, Leicester, UK
Corresponding author: Mosbah Alown (e-mail: mosbah.alown@dmu.ac.uk).

ABSTRACT Electronic voting (e-voting) systems have significantly improved the traditional voting process by addressing key concerns such as security, public acceptability, and convenience. However, these systems often face unique challenges, such as ensuring voter privacy and verifiability, preventing coercion and double voting, and maintaining scalability while protecting participant confidentiality. This study critically analyses and compares various e-voting schemes and technologies, evaluating their security features, verifiability mechanisms, and potential vulnerabilities. This paper reviews Direct Recording Electronic (DRE) voting, internet voting, and blockchain-based e-voting systems. In so doing, we provide an understanding of cryptographic primitives employed in e-voting systems and how they address specific characteristics and challenges associated with each voting scheme. Furthermore, we examine the applications proposed by previous studies in the context of these voting systems, assessing their strengths, limitations, and impact on democratic procedures. The cryptographic primitives reviewed include techniques like homomorphic encryption, blind signatures, and zero-knowledge proofs, which can enhance voter privacy, verifiability, and resistance to coercion and double voting.

INDEX TERMS Electronic voting, Internet voting, Blockchain, Decentralised Ledger, Security, Privacy.

I. INTRODUCTION

TRADITIONAL paper-based voting systems, while historically standard, face fundamental challenges that threaten democratic processes. These systems are vulnerable to human error in vote counting, physical tampering, and fraud, which can compromise election integrity and diminish public trust [1]. Moreover, the manual nature of paper-based voting creates functional limitations in voter registration, ballot allocation, and vote tallying, leading to increased costs and delayed results [2]. The digital revolution has started changing toward electronic voting (e-voting) systems as a potential solution to these challenges. E-voting promises to enhance accessibility by enabling remote participation while potentially improving accuracy through automated counting [3]. However, e-voting systems face cryptographic challenges in providing voter anonymity to preserve privacy while ensuring the accuracy and legitimacy of votes [4]. The success of e-voting depends on addressing these challenges to build public trust and ensure widespread acceptance of election outcomes. The introduction of DRE voting systems, Internet-based voting, and the rise of blockchain technology Have transformed how societies shape their governments' policies. DRE systems can offer convenience in the ballot-casting process by eliminating the limitations of paper-based methods [5]. Internet-based voting further enhances this convenience by allowing individuals to vote anywhere worldwide. Simultaneously, blockchain technology's emergence and growing adoption has generated interest in using it for e-voting [6]. Blockchain technology offers unique advantages that directly address the core challenges of electronic voting. Its immutable, decentralised architecture provides a transparent yet tamper-resistant record of all transactions [7]. This inherent security, combined with the technology's success in other sensitive domains like healthcare and finance, positions blockchain as a potential cornerstone for secure, transparent, and trustworthy e-voting systems. However, for these security benefits to spread acceptance, voters must understand how this technology protects their votes. Educating the public on the security and reliability of blockchain is key to building trust, which is essential for people to accept and adopt blockchain-based voting systems widely. We aim to comprehensively assess how different electronic voting approaches fulfil the core principles of democratic elections. This includes evaluating their effectiveness in ensuring voter privacy, double voting prevention, coercion mitigation, and achieving verifiability. Additionally, the research addresses the technical challenges inherent to these systems, offering a critical analysis of their capabilities to meet these requirements within the context of evolving technological and security landscapes.

A. RESEARCH METHODOLOGY

This study employs a survey-based research methodology to analyse the current state of e-voting systems. The research comprises a comprehensive literature review of academic papers, technical reports, and case studies on e-voting implementations. Data is collected through secondary sources and categorised based on the DRE, Internet and blockchain protocols used, security measures, and challenges faced. The study uses thematic analysis to compare solutions, identify common challenges, and suggest improvements.

B. OUR CONTRIBUTIONS

Our work aims to provide a broader understanding of e-voting, focusing on security and privacy frameworks. In addition to the survey, we conducted research on various aspects of e-voting to support our findings, including:

1) An extensive, up-to-date review to understand the key cryptographic principles for e-voting systems. As such, this study focused on how these principles meet contemporary voting needs.
2) A critical analysis of blockchain technology in the context of e-voting. We emphasise blockchain technology's significance and wide-ranging impact in advancing blockchain-based voting systems.
3) An in-depth analysis of modern e-voting machines, including a detailed comparison and comprehensive evaluation. Our study examines the functionality of these systems and their methods of handling voting and counting and identifies existing gaps.
4) An exploration of the security features of different e-voting systems.

C. ROADMAP

The sections of this survey are organised as follows: Firstly, Section II provides an overview of Cryptography in e-voting and further explains concepts and principles that underpin e-voting systems in general. Then, Section III provides background for both Blockchain and Distributed Ledger Technology, and by doing so, it emphasises the role of blockchain in improving e-voting systems. Moreover, section IV explores Layer 2 solutions to address blockchain scalability challenges, highlighting recent research advancements. Further, Section V unpacks the Model for Security and Privacy of e-voting systems while explaining frameworks and considerations employed in practice to maintain the security of the voting process. Going forward, Section VI outlines this work by analysing previous studies and highlighting their limitations and contributions to e-voting systems. Moreover, in Section VII, we present a detailed comparison between the studies above, and in so doing, we identify correlations, differences, and trends among them. Finally, Section IX concludes the survey by summarising the main empirical findings of this research while discussing its implications for future investigation in e-voting security.

II. CRYPTOGRAPHY BACKGROUND

This section explains and highlights the applications of Public Key Infrastructure (PKI), Blind Signatures, Homomorphic Encryption, Zero-Knowledge Proofs, Mix-nets, Hash Functions, and Digital Signatures. These techniques are important to ensure both security and privacy in e-voting. Here, we thoroughly examine their theoretical foundations as well as how they are implemented in practice. By understanding these tools, we can better evaluate the effectiveness of different e-voting systems.

A. CRYPTOGRAPHIC HASH FUNCTIONS

Cryptographic hash functions are important for ensuring data integrity and security. They convert input data into a fixed-length string of bits, known as digests or hash codes [8]. A Hash function generally needs to be pre-image resistant, second-image resistant, and collision resistant to be cryptographically secure. Within the hash space, the hashes are generally evenly distributed. In other words, if someone tries to guess which hash would be created by a given input, that person is less likely to have a probability more than that of random guessing. Moreover, a hash function could also be a compression function (which produces a shorter string from a fixed-length input string). $f : \{0,1\}^{m+t} \rightarrow \{0,1\}^m$. Furthermore, a cryptographic hash function must also be resistant to all known cryptanalytic attacks. As such, at least, it must possess the following characteristics:

For a given $h : X \rightarrow Y$, we state that h is:

- pre-image resistant (one-way). if given $y \in Y$ it is computationally infeasible to find a value $x \in X$ such that $h(x) = y$.
- 2-nd pre-image resistant (weak collision resistant). Finding a value $x' \in X$, such that $x' \neq x$ and $h(x^{'}) = h(x)$ is computationally infeasible.
- collision resistant. Finding two different values $x', x \in X$, such that $h(x') = h(x)$ is computationally infeasible.

Cryptographic hash functions are fundamental to the integrity and security of blockchain technology as they store the hash value of the last block in the chain in the current block and also enable links between blocks back to the genesis block (e.g., SHA256 and Keccak (SHA3)) [8].

B. PUBLIC KEY INFRASTRUCTURE (PKI)

PKI ensures secure communication in e-voting by confirming public keys and building reliable networks [9]. This emphasises its extensive range of uses and its key role in maintaining e-voting integrity. To achieve robust security foundations, PKI relies on many cryptographic algorithms, among which are RSA (Rivest-Shamir-Adleman) [10], DSA (Digital Signature Algorithm) [11], ECDSA (Elliptic Curve Digital Signature Algorithm) [12], and ElGamal [13]. These algorithms contribute to the secure operation of PKI systems and ensure the integrity and confidentiality of all electronic communications. RSA factorises large integers to generate public and private key pairs. On the other hand, DSA utilises modular exponentiation and discrete logarithm problems to provide secure digital signatures. Moreover, ECDSA offers a more efficient alternative for digital signature generation and verification as it is based on elliptic curve mathematics. Finally, ElGamal is a probabilistic encryption algorithm that secures data transmission using discrete logarithm problems. These PKI algorithms and their benefits show the flexibility and effectiveness of PKI in strengthening security frameworks, and they can be used for various applications, including e-voting systems.

C. DIGITAL SIGNATURES

A digital signature is becoming a new alternative to handwritten signatures because it assures a document's origin, further protecting it against impersonation [14]. It thus authenticates electronically transmitted messages and documents while, at the same time, it ensures security and also verifies all parties involved.

The process of creating a digital signature involves the following steps [15]: Let $\mathcal{M}$ represent the set of possible messages or files, $\mathcal{K}$ denote the set of possible private keys, and $\mathcal{V}$ denote the set of possible public keys.

KEY GENERATION (KEYGEN)

This algorithm generates the sender's public key pk and private key sk as follows:

$$KeyGen() \rightarrow (pk, sk)$$

SIGNING (SIGN)

Given a message or file $m \in \mathcal{M}$ and the sender's private key $sk \in \mathcal{K}$, the signing algorithm generates a digital signature s as such:

1) Apply a hash function H to the message m to compute a unique hash value $h = H(m)$.
2) Encrypt the hash value h using the sender's private key sk to obtain the signature $s = Enc_{sk}(h)$.
3) Attach the signature s to the message m as the digitally signed message $m_s = (m, s)$.
4) Transmit the digitally signed message m_s to the recipient.

VERIFICATION (VERIFY)

To verify the validity of a digital signature, the verifier follows these steps:

Given a digitally signed message $m_s = (m, s)$ and the sender's public key $pk \in \mathcal{V}$:

1) Retrieve the signature s and the corresponding message m from the digitally signed message m_s.
2) Utilise the sender's public key pk to decrypt the signature s and obtain the original hash value $h = Dec_{pk}(s)$.
3) Apply the same hash function H to the received message m to compute a new hash value $h' = H(m)$.
4) Compare the retrieved hash value h with the newly computed hash value h'.
5) If $h = h'$, consider the digital signature as valid.
6) If $h \neq h'$, consider the digital signature as invalid.

D. BLIND SIGNATURES

Blind signatures, a specialised form of digital signatures, enable a signer to authenticate a message without knowledge of its content [16]. This cryptographic technique ensures that the signer cannot later deny or trace their signature. In e-voting, a voter asks authorised users to sign a hidden ballot to cast his/her vote anonymously later [17]. A blind signature operates under the assumption that the requester interacts with the signer to get a signature on documents where he/she can conceal the document's information. If voters cannot locate their vote in the final tally, they can produce evidence of the authority's signature on their blind ballot without compromising vote secrecy. As such, all anonymous votes will be revealed during the subsequent tallying phase.

This blind signature scheme consists of five distinct algorithms: Key Generation, Blinding, Signing, Unblinding, and Verification [18].

A blind signature scheme is a cryptographic protocol that involves two parties: a signer and a user. The scheme consists of these algorithms:

1) **Key Generation (*KeyGen*):** This algorithm generates the signer's public key pk and private key sk. These keys are used for blind signing operations.
$KeyGen() \rightarrow (pk, sk)$
2) **Blinding (*Blind*):** The user blinds the message m using the signer's public key pk. This process involves a random blinding factor r, and the result is a blinded message m_b.
$Blind(pk, m, r) \rightarrow m_b$
3) **Signing (*Sign*):** The signer takes the blinded message m_b and produces a blinded signature s_b.
$Sign(sk, m_b) \rightarrow s_b$
4) **Unblinding (*Unblind*):** The user unblinds the blinded signature s_b using the inverse of the blinding factor r, resulting in the actual signature s.
$Unblind(pk, s_b, r) \rightarrow s$
5) **Verification (*Verify*):** The validity of the signature s is verified using the signer's public key pk and the original message m.
$Verify(pk, m, s) \rightarrow$ True/False

The blind signature scheme ensures that the signer cannot link the blinded message m_b to the actual message m, which helps preserve the user's privacy. Further, the user receives a valid signature s for the original message m despite the signer remaining unaware of the actual content being signed. This property is key for applications where privacy is prioritised, which includes applications such as e-voting or cryptographic protocols (that involve anonymous authentication). Blind signatures are essentially a widely used cryptographic technique which can be seen in e-voting. They have been successfully implemented in schemes such as RSA [19], Schnorr [20], DSA [21], and ECDSA [22]. These schemes have been extensively examined but also applied in the context of blind signatures, and this ensured that voters' privacy and anonymity are always prioritised while obtaining valid signatures on sensitive documents or transactions.

E. HOMOMORPHIC ENCRYPTION

Homomorphic encryption enables users to do calculations on the ciphertext without decrypting [23]. The homomorphic characteristic can be used to make a secure e-voting system that retrieves data with high anonymity. A homomorphic encryption can be either additive homomorphic $\oplus$ or multiplicative homomorphic $\otimes$ [24]. $\mathrm{Enc}(m_1 \oplus m_2)$ and $\mathrm{Enc}(m_1 \otimes m_2)$, for instance, can be derived from $\mathrm{Enc}(m_1)$ and $\mathrm{Enc}(m_2)$, respectively.

Let $\mathcal{P}$ denote the set of all plaintexts, $\mathcal{C}$ denote the set of all ciphertexts, and $\mathcal{K}$ represent the set of possible encryption keys.

A homomorphic encryption scheme is a triple of algorithms, denoted as $(\mathrm{KeyGen}, \mathrm{Enc}, \mathrm{Dec})$, satisfying the following properties:

1) $\mathrm{KeyGen}(1^\lambda) \rightarrow \mathcal{K}$: The key generation algorithm, where 1^λ is the security parameter, outputs an encryption key $pk \in \mathcal{K}$ with respect to the security parameter λ.
2) $\mathrm{Enc}(pk, m) \rightarrow \mathcal{C}$: The encryption algorithm takes as input the encryption key pk and a plaintext message $m \in \mathcal{P}$ and outputs a ciphertext $c \in \mathcal{C}$.
3) $\mathrm{Dec}(sk, c) \rightarrow \mathcal{P}$: The decryption algorithm takes as input the secret key sk and a ciphertext $c \in \mathcal{C}$ and outputs the corresponding plaintext message $m \in \mathcal{P}$.

Furthermore, a homomorphic encryption scheme should satisfy the following homomorphism property:

4) For any $m_1, m_2 \in \mathcal{P}$ and their corresponding ciphertexts $c_1 = \mathrm{Enc}(pk, m_1)$ and $c_2 = \mathrm{Enc}(pk, m_2)$, the homomorphic property holds for the homomorphic operation f:
$\mathrm{Dec}(sk, f(c_1, c_2)) = f(m_1, m_2)$
Where f is a function that can be computed efficiently in the plaintext space.

To illustrate more, a homomorphic encryption scheme helps perform some algebraic operations (e.g., addition, multiplication) on encrypted ciphertexts, which can result in the same operations being performed on the corresponding plaintexts when decrypted. This property allows for computations to be carried out on encrypted data without revealing the actual plaintexts, thus preserving privacy while still obtaining valuable results. Moreover, Fully Homomorphic Encryption (FHE) is used for arbitrary computations on encrypted data, while Partially Homomorphic Encryption (PHE) supports either addition or multiplication operations on encrypted data.

F. ZERO-KNOWLEDGE PROOFS (ZKPS)

ZKP is also a cryptographic technique utilised to test the correctness of a message without disclosing extraneous details [25]. Proof must be zero-knowledge in the context of e-voting for various reasons. For example, a voter might confirm that his/her votes are being counted, but they will not get any more information to show their voice buyer. Another example is that a voter may verify that his/her votes are being tallied, but they will not receive any additional information exposing their choice to a third party. In a blind signature-based technique, the voter must always provide evidence that the blind vote is valid and conforms to a predetermined format. Furthermore, integrating ZKP into an e-voting system based on blockchain technology enables individuals to verify their identity or other confidential information without disclosing the actual data. This specific approach is highly efficient in ensuring the privacy and security of votes.

1) Σ Proofs

Let $\mathcal{R}$ represent a relation, where for a statement x, $R(x) = 1$ indicates that the statement is true, and $R(x) = 0$ indicates that it is false. A Σ-proof scheme is a protocol involving a prover and a verifier. It allows the prover to convince the verifier of the truth of a statement without revealing any further information (beyond whether it is true or not). The scheme consists of the following algorithms:

- **Key Generation (*KeyGen*):** This algorithm generates a prover's public key pk and private key sk.

$$KeyGen() \rightarrow (pk, sk)$$

- **Proof Generation (*Prove*):** Given a statement x, the prover generates a proof π to convince the verifier of the truth of x without revealing x. The proof π is generated as follows:

$$Prove(sk, x) \rightarrow \pi$$

- **Proof Verification (*Verify*):** The verifier takes the prover's public key pk, the statement x, and the proof π as input. The verifier checks whether the proof is valid and whether it convinces the verifier that $R(x) = 1$.

$$Verify(pk, x, \pi) \rightarrow \text{Accept/Reject}$$

The zero-knowledge property is achieved when the scheme satisfies the following:

- **Completeness:** If $R(x) = 1$, an honest prover can convince the verifier with high probability.
- **Soundness:** A dishonest prover cannot convince the verifier of a false statement except with negligible probability.

- **Zero-Knowledge:** The verifier learns nothing beyond the truth or falsity of the statement.

The actual size of ZKP ultimately constrains its application. This is because due to the requirement of including these proofs in the blockchain, their size should be minimal.

2) zkSNARKs

ZKSNARKs (Zero-Knowledge Succinct Non-Interactive Arguments of Knowledge) are cryptographic proof systems that efficiently verify the correctness of nonlinear functions without revealing any underlying data [26]. zkSNARK, as previously mentioned, is a cryptographic protocol that allows a prover to convince a verifier that they have certain knowledge (a witness) about a statement without revealing who the witness is. zkSNARK is a new version of zero-knowledge cryptographic proofs [27]. It is probably the most desirable proving system for the verifier because of its small fixed proof size and fixed interval verification costs, especially when compared to the traditional ZKP (even for arbitrarily huge relations). In zkSNARKs, the prover constructs a proof π using public input/output data (statement ϕ) and private input data (witness w) corresponding to a designated function. Following that, a verifier can find out the validity of the statement ϕ by checking the proof π, without the need to access the private input data w. The zkSNARK scheme involves these components:

- **Circuit:** A circuit $\mathcal{C}$ represents a computation or a statement that the prover wants to prove knowledge of. The circuit consists of logical gates and arithmetic operations.
- **Witness:** A witness w is the secret information the prover possesses and wants to prove knowledge. The witness corresponds to the input values that satisfy the circuit's constraints.
- **Setup (*Setup*):** The setup algorithm generates public parameters that are used by both the prover and verifier to perform zkSNARK operations.

$$\mathit{Setup}(1^{\lambda}) \rightarrow \text{Public parameters}$$

- **Proving (*Prove*):** Given the public parameters, a circuit $\mathcal{C}$, and a witness w, the prover generates a zkSNARK proof π that convinces the verifier of the validity of the witness's claim regarding the circuit.

$$\mathit{Prove}(\text{Public parameters}, \mathcal{C}, w) \rightarrow \pi$$

- **Verification (*Verify*):** The verifier takes the public parameters, the circuit $\mathcal{C}$, the zkSNARK proof π, and a statement *stmt* as input. The verifier checks whether the proof is valid and whether the statement is true based on the proof.

$$\mathit{Verify}(\text{Public parameters}, \mathcal{C}, \mathit{stmt}, \pi) \rightarrow \text{True/False}$$

In addition to the properties of completeness, soundness, and zero-knowledge already discussed in the context of ZKPs, ZKSNARKs also possess the following key properties [28]:

- **Succinct:** The proof size must be very small to allow verification within a few milliseconds.
- **Non-Interactive:** Non-Interactive: The prover sends a single set of information to the verifier for verification without any interaction between them.
- **Argument:** A proof is computationally sound if it remains valid against a prover with limited computing power. This means the proof works when the prover can only do calculations in a reasonable amount of time, not with unlimited resources.
- **Of Knowledge:** The proof shows that the prover knows a secret that needs to be proven in the statement. Without this secret, the prover cannot provide valid proof.

3) zkSTARK

A zkSTARK (Zero-Knowledge Scalable Transparent Argument of Knowledge) is a cryptographic protocol that allows a prover to convince a verifier of the validity of a statement while keeping the underlying data and computation fully private [29]. zkSTARK schemes achieve this by generating proof that the verifier can prove without interacting with the prover. The zkSTARK scheme involves these components:

- **Statement:** A statement *stmt* represents a claim that the prover aims to demonstrate. It could be a mathematical proposition, a computation, or other assertion.
- **Witness:** A witness w is the private information the prover holds and uses to prove the statement's validity *stmt*.
- **Polynomial Constraints (PC):** PCs are essential to zkSTARK protocols, linking input data, witness, and output data of computations. There are several zkSTARK protocols, each using different proof constructions. As explained in [30], these constraints verify computational correctness while maintaining data privacy. Recent implementations, such as Plonky2 [31] and RISC Zero [32], demonstrate various PC applications. Plonky2 enhances proof system efficiency, while RISC Zero utilises PCs in a zero-knowledge virtual machine for general computations.
- **Setup (*Setup*):** The setup algorithm generates public parameters that are used by both the prover and verifier to perform zkSTARK operations.

$$\mathit{Setup}(1^{\lambda}) \rightarrow \text{Public parameters}$$

- **Proving (*Prove*):** Given the public parameters, a statement *stmt*, a witness w, and the polynomial constraints, the prover generates a zkSTARK proof π that confirms the validity of *stmt* without revealing w.

$$\mathit{Prove}(\text{Public parameters}, \mathit{stmt}, w, \text{PC}) \rightarrow \pi$$

- **Verification (*Verify*):** The verifier takes the public parameters, the statement *stmt*, and the zkSTARK proof π

as input. The verifier checks whether the proof is valid and whether the statement is true based on the proof.

$$Verify(\text{Public parameters}, stmt, \pi) \rightarrow \text{True/False}$$

zkSTARK is a knowledge system argument for an NP-complete computational integrity relation. According to [30] and In addition to the properties of completeness, soundness, and zero-knowledge already discussed in the context of ZKPs, it has the following features:

- **Non-interactivity:** The proof can be generated without back-and-forth communication between the prover and verifier.
- **Scalability:** The system achieves a polylogarithmic proof size and verifier time complexity, coupled with quasilinear prover time complexity.
- **Transparency:** All randomness utilised in the setup phase is public, ensuring full process transparency.

G. SNARK-FRIENDLY HASH FUNCTIONS

The current academic research in this field is focused on developing novel SNARK-friendly cryptographic tools to reduce the computational overhead of zero-knowledge proofs. This is mainly used in the context of hash functions [33]. SNARK-friendly hash functions are designed to be more efficient within zero-knowledge proof systems because they impose fewer computational constraints. For example, traditional hash functions like SHA have complex structures that make them computationally expensive in ZKSNARKs. In contrast, SNARK-friendly hash functions are optimised to minimise these constraints, making them more suitable for privacy-centric applications like blockchains, where it is crucial to demonstrate knowledge of information without revealing it. Key examples of such hash functions include Poseidon [34], MiMC [35], and Sinsemilla [36]. Research projects like Semaphore [37] highlight the successful integration of these hash functions into blockchain protocols, representing a significant advancement in the secure and efficient application of ZKPs.

H. MIX-NETS

A mix-net (or a re-encryption mix-net) function is to re-encrypt the ciphertext more than once over a collection of ciphertexts while randomly rearranging their order. Using this re-encryption, a shuffle agent would modify the encrypted text into another number without necessarily decrypting it [38] first, leaving the original ciphertext's decryption unaffected. By cascading several shuffle agents, the authority could not recognise the ciphertext's initial order, making it difficult to verify its originality.

Let $\mathcal{M}$ represent the set of plaintext messages and $\mathcal{C}$ denote the set of ciphertexts. A re-encryption mix-net involves the following components:

- **Ciphertext Encryption:** Given a plaintext $m \in \mathcal{M}$, an encryption algorithm Enc produces a ciphertext $c = \text{Enc}(m)$.
- **Re-encryption** (*ReEnc*)**:** The re-encryption algorithm takes as input a ciphertext c_i and re-encrypts it to produce a new ciphertext c_j without revealing the plaintext contents.

$$ReEnc(c_i) \rightarrow c_j$$

- **Shuffling** (*Shuffle*)**:** The shuffle agent randomly rearranges the order of the ciphertexts while maintaining the integrity of the encryption.

$$Shuffle(c_1, c_2, ..., c_n) \rightarrow c_{\pi(1)}, c_{\pi(2)}, ..., c_{\pi(n)}$$

Essentially, a re-encryption mix-net functions by iteratively employing the *ReEnc* algorithm on ciphertexts, resulting in a sequence of encrypted data. The shuffle agent subsequently rearranges these ciphertexts to mask the original order. This process achieves two critical properties:

- **Privacy:** The re-encryption mix-net ensures that the original plaintexts remain confidential even when the ciphertexts undergo multiple transformations.
- **Anonymity:** The Shuffling of ciphertexts prevents adversaries from linking the input ciphertexts with the corresponding output ciphertexts. This further enhances the anonymity of communication.

The use of re-encryption and shuffling collectively protects the content and source of messages, thus raising the importance of re-encryption mix-nets for achieving more secure and private communications.

III. BLOCKCHAIN AND DISTRIBUTED LEDGER TECHNOLOGY

Blockchain technology consists of a series of time-stamped and linked cryptographic hashes. They then create an unbreakable chain of records [39]. Every new block that gets added maintains the hash of the previous block's data, which leads to a constant expansion of the chain as new blocks are added. This concept is the foundation for a secure and universally accessible data repository and finding applications in cryptocurrencies, related industries, and various transaction-oriented sectors.

A. TYPES OF BLOCKCHAIN

The technology's applications are categorised into different types of blockchains:

1) Public Blockchain:

This type is open to all users without the need for special authorisation [40]. It is accessible for anyone to read, write, and contribute. Furthermore, public blockchains operate decentralised, where no single entity controls the network. This makes the data accessible to the public, irreversible, and secure.

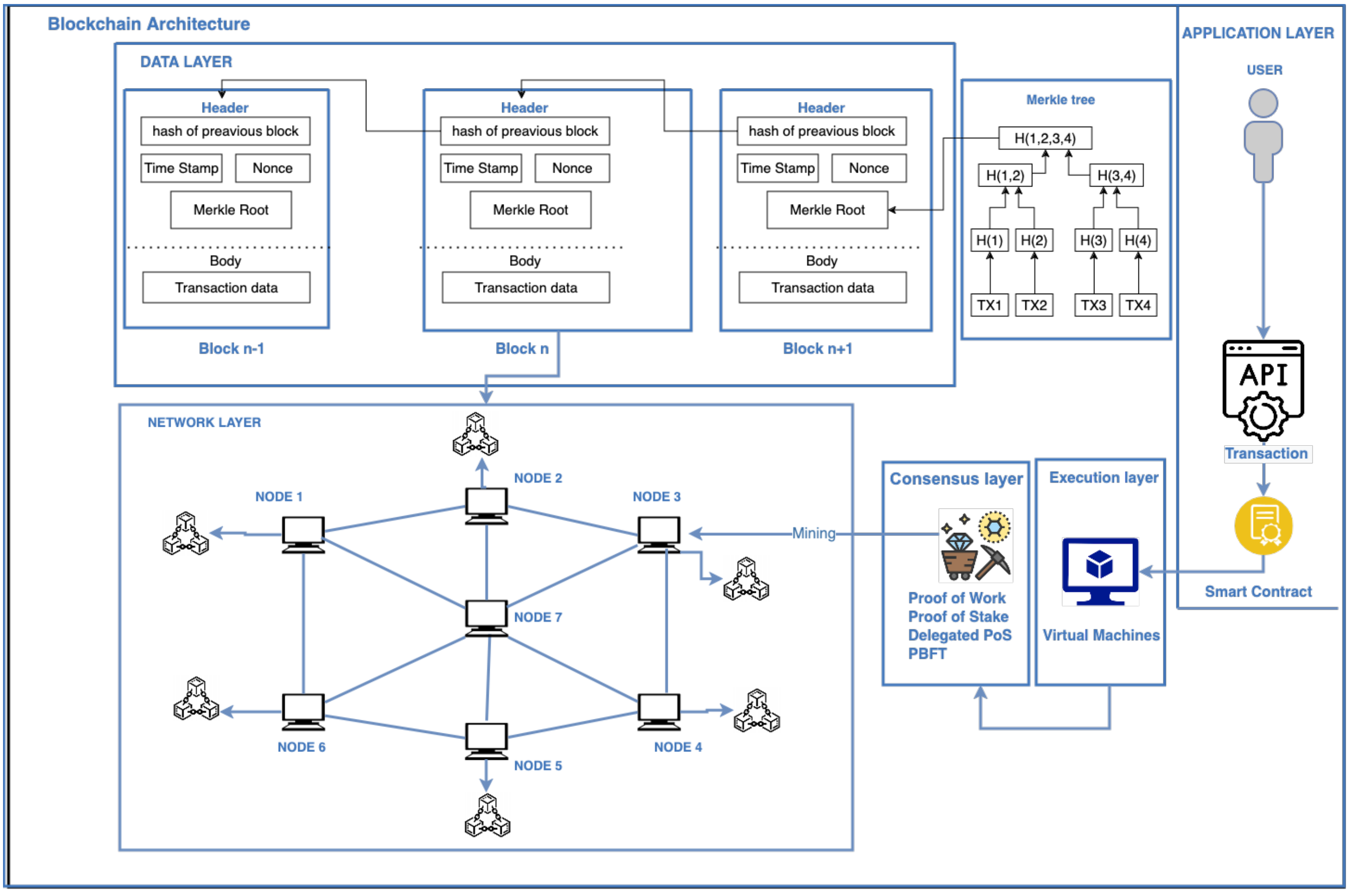

FIGURE 1. ***Overview of the High-Level Blockchain architecture. The figure portrays the layered components of a blockchain system, encompassing Application, Execution, Consensus, Network, and Data layers.***

2) Private Blockchain:

This is referred to as permissioned blockchains. This type limits consensus participation and access to a specific group of peers with authorised rights [41]. In this type, write permissions are concentrated in a defined group and restrict network accessibility.

3) Consortium Blockchain:

The consortium blockchain merges the benefits of both public and private blockchains [42]. Instead of requiring all network nodes to execute the consensus algorithm, it employs preselected and authorised nodes, reducing network overhead. Managed by multiple organisations and operating on permission principles, it resembles private blockchains but is decentralised, ensuring reliable data transactions and suitable applications. These classifications show the diverse applications of blockchain technology, making it both flexible and, ultimately, a cornerstone of modern decentralised systems.

B. LAYERS OF BLOCKCHAIN

Different layers of architectures and frameworks characterise the blockchain, as they vary depending on the specific use case and type of blockchain. Previous research has described these design variations and presented their diverse approaches and models [43]–[46]. For this study, we have adopted a common framework that provides a comprehensive understanding of the layers involved in blockchain technology and allows for meaningful comparison across different research works.

Figure 1 illustrates the structure of the High-Level Blockchain architecture, this outlines the fundamental components of a Blockchain system. These components play an important role in ensuring data security and integrity. As mentioned previously, blockchain technology is characterised by a multi-layered architecture that is used to specify its main operational framework. The Application Layer is the first point for users where they initiate critical transactions within the blockchain ecosystem. After that, the Execution Layer encompasses Smart Contracts, autonomous scripts capable of executing predetermined actions within decentralised applications. Then, the Consensus Layer employs a range of mechanisms, including PoW, PoS, DPoS, and PBFT, to arrange consensus and ensure uniform data coherence among network participants. In the Network Layer, a complex network of Nodes maintains synchronised copies of blockchain data collaboratively, enabling data sharing, redundancy, and synchronisation. The core of this architecture is in the Data Layer, where Blocks are structured into a Merkle Tree configuration. Each block has a sequence of transactions, and the Merkle

Tree structure enhances data integrity and verification.

1) Application/Presentation layer

The application layer includes blockchain programs and apps, playing a key role in how blockchain technology works [47]. In its first stages, this layer focused mainly on enabling the transfer of cryptocurrencies, a key function introduced by Bitcoin and its several iterations. Nevertheless, adding smart contracts greatly expanded what it can do, now covering decentralised platforms for supply chain management, identity verification, and notarial services [48]. Its main goal is to provide a user-friendly interface that eliminates the way in which different applications interact with the leading blockchain network. As such, the application layer makes blockchain technology accessible to end-users, software developers, and commercial companies. The blockchain application layer supports a complex framework that powers the blockchain's capabilities. Some of the most crucial elements that make this framework operational include:

a: Smart Contracts

In the blockchain application layer, *smart contracts* serve as algorithmic agreements that are automatically executed upon the fulfilment of predefined conditions [49]. Once all parties involved have signed, the digital agreements are integrated with encoded commands and transformed into programming code structures. Bitcoin transactions demonstrate how contractual codes can be added to the blockchain, spread across the peer-to-peer (*P2P*) network, and then validated by network nodes [50]. Smart contracts have predetermined states and transition rules, indicating their dynamism. These templates cover various circumstances and are crucial in initiating contract activation through timed intervals or specific events. The blockchain functions as a decentralised, immutable ledger that actively monitors the real-time status of smart contracts. It executes contracts after specific trigger conditions are satisfied, following the principles of "if-then" logic [51]. This approach differs from traditional contractual paradigms by shifting from manual to algorithmic enforcement. The combination of smart contracts and blockchain technology facilitates self-executing agreements resistant to intermediaries, underpinned by principles of trust, verifiability, and autonomy.

b: Decentralised applications (DApps)

DApps share many qualities with traditional applications. However, they are often called "trustless" or "peer-to-peer" due to their lack of centralised control and facilitation by individual servers or entities. The main difference comes from Blockchain technology, which provides the computing power and data needed to run without central oversight [52]. Unlike traditional apps with concentrated control, DApps exhibit a unique operational approach based on decentralised principles. A DApp can be seen as a website combined with one or more smart contracts, which set the unchangeable parts of the app's operation. A front-end application is essential to comprehensively understand the operating environment of a DApp, as it allows end users to interact with the system, manipulate state variables, and execute functions within smart contracts [53]. The division of responsibilities shows the interdependent connections between the front-end and smart contracts, working together to ensure smooth operation and user satisfaction in DApps.

c: Wallets

Digital wallets have the potential to entirely change the e-voting landscape [54]. These wallets provide users with secure access to their digital transactions and incorporate public and private keys, along with integrated security mechanisms. By integrating voting capabilities into digital wallets, users can engage in various voting procedures directly, enhancing the efficiency, transparency, accountability, and security of the e-voting system. This integration eliminates the need for physical ballots and allows for the secure recording of voting data onto the blockchain, guaranteeing transparency and immutability. The use of digital wallets in e-voting presents a new model of governance that incorporates the principles of trust, decentralisation, and verifiability, transforming the traditional voting landscape.

2) Execution/Infrastructure layer

The execution layer executes running contracts or low-level machine code (bytecode) in a runtime environment installed on each Blockchain network node [55]. A transaction is executed within this processing setup.

a: Virtual Machine (VM)

VMs enable operating system virtualisation [56]. They provide users access to complete operating systems, allowing them to run diverse application packages and emulate different devices within the cloud environment. Active nodes store and execute these virtual machines in a network to process incoming commands. In Ethereum, smart contracts are written in code with if-when statements. When the conditions are met, the smart contract executes the agreed terms [57]. A transaction starts the contract, is processed by an Ethereum node and then passed to a VM. This VM runs the contract on the blockchain, allowing all contributors to see updates. The contract code is shared among all contributors without a central authority controlling it. The blockchain lets participants agree on or modify the contract through their access. For e-voting, VMs and blockchain can create a secure and transparent system. E-voting applications in VMs ensure that each instance runs independently and securely. Using the Ethereum Virtual Machine (EVM) to run smart contracts makes the voting process transparent and tamper-proof [58]. This combination offers flexibility, scalability, and efficient resource use, enhancing security and performance in the voting process.

b: Container

A container is an independent, self-contained package to execute a program or service [59]. It contains the application files, software libraries, and hardware requirements in one component. A container is a concept similar to virtual machines, but it is more widely used due to the lightness of its size and speed compared to a VM. In addition, the application's portability feature allows it to be executed from anywhere and has easy scalability. As an open platform, Docker accelerates software delivery by separating applications from infrastructure, offering features such as rapid code testing, modular architecture updates, and comprehensive development tools, including graphical interfaces, command lines, and APIs [60]. It establishes a constant runtime environment for applications across various hardware and operating systems.

3) Consensus layer

Consensus algorithms enable a distributed or decentralised network to reach decisions swiftly and unanimously whenever required [61]. So even if some peers fail, the network of peers stays reliable for sharing information. Its characteristics include guaranteeing decentralised governance, a quorum structure, authentication, integrity, non-repudiation, and performance. Although numerous consensus algorithms exist, most research in the consensus layer focuses on enhancing key algorithms such as Proof of Work (PoW), Proof of Stake (PoS), Delegated Proof of Stake (DPoS) and Practical Byzantine Fault Tolerance (PBFT) [62]–[66].

a: Proof of work (PoW)

The PoW consensus algorithm is often used in public blockchains like Bitcoin. It involves nodes competing to solve complex computational puzzles to verify transactions and generate new blocks [67]. The node that successfully solves the challenge is granted the opportunity to append the following block to the chain and rewarded accordingly. The main goal of *PoW* is to establish consensus among nodes regarding the current state of the blockchain. Nodes expend extensive computational resources to find a solution; this collective effort is a substitute for reaching a consensus. The probability of a node's solution being accepted is directly proportional to the processing power they have given. To breach the integrity of a PoW blockchain, an adversary would need to control more than 50% of the network's overall computing power, which would require an extraordinarily resource-intensive and economically infeasible for large, established networks like Bitcoin. as long as the network remains sufficiently decentralised, PoW blockchains are highly secure against 51% attacks.

b: Proof of stake (PoS)

PoS is a consensus method developed as a less energy-intensive alternative to PoW protocols [68]. In PoS, nodes in the blockchain system choose to produce blocks in a deterministic or pseudo-random fashion, with the likelihood of their selection associated with their wealth or stake. PoS differs from PoW by eliminating the requirement for miners to allocate computational resources to elect a leader [69]. Instead, miners participate in a procedure where a participant is chosen randomly, with the likelihood of selection directly related to their stake, as shown in the existing blockchain ledger. This approach introduces a self-referential system within the blockchain, where the preservation of the blockchain relies on the stakeholders, who allocate responsibilities and rewards according to their respective stake holdings. While a proof-of-stake protocol holds potential, its implementation presents challenges in definition, technical aspects, and analysis.

c: Delegated Proof of Stake (DPoS)

The *DPoS* consensus mechanism enhances the *PoS* system, providing a more efficient approach to transaction validation compared to the traditional *PoW* methodology [70]. In *DPoS*, transaction validators are chosen via a voting procedure, speeding up block production and improving energy efficiency [69]. However, DPoS has limitations, including reduced decentralisation and potential security vulnerabilities. In DPoS-based blockchain networks, a specific group of witnesses or delegates, selected based on their interests and investments, is responsible for validating transactions. The rotating methodology employed by DPoS enhances transaction confirmation efficiency and promotes resource preservation, distinguishing it from PoW and PoS systems.

d: Practical Byzantine Fault Tolerance (PBFT)

Byzantine Fault Tolerance (BFT) and its instantiation, PBFT, are foundational concepts within consensus algorithms designed for distributed systems that deal with malicious or defective nodes. PBFT, introduced by [71], represents an evolution of BFT by addressing practical feasibility and operational efficiency. PBFT works in distinct phases and requires a fixed number of nodes, able to handle a maximum of one-third as faulty, requiring the participation of $3f + 1$ nodes for reliable operation (f denoting the maximum possible faulty nodes). While PBFT ensures the toleration of f Byzantine nodes, it comes with a significant communication complexity, which increases in proportion to the square of the number of nodes.

4) Network layer

The network layer (P2P) is responsible for transmission between nodes. The network layer controls various data transmission mechanisms and verification methods, including discovery, transactions, and block propagation [72]. Distributed nodes in a computer network divide up tasks and work together towards a common objective in what is known as a peer-to-peer network. In order to keep the Blockchain network running smoothly, this P2P layer ensures that nodes can find one another and communicate, distribute data, and remain synchronised.

5) Data Layer

The data layer ensures data integrity using the block data structures [73]. This layer covers digital signature, chain constructing, hashing, Merkle tree, time stamp, and other techniques. As the network expands, every node produces a new block of information connected to the longest primary blockchain and contains all of the transactions it has received during that period.

a: Transactions

The transaction involves transferring an x value from person A to person B [74]. When a transaction is created, it is broadcast to all network nodes. Consequently, each Node will receive and process the transactions. Within each transaction, the Node will trace the origin of the components to ensure their legitimacy and that they are in the possession of the intended recipient. By utilising digital signatures, the system ensures that only the intended recipients of the transaction or data can access, read, and process it. Furthermore, the digital signatures verify the authenticity of the participants and the transaction itself.

b: Merkle tree

Merkle trees are binary trees with many leaf nodes at the bottom and store a set of intermediate nodes [75]. Each node is the hash of its two children, and a single root node, also derived from its two children's hash, points to the "head" of the tree. With the Merkle tree, data in a block can be given in pieces while still being guaranteed accuracy. A node can download just a block's header from one source and the portion of the Merkle tree that applies to them from another source. Suppose an adversary user tries to insert a phoney transaction into the root of a Merkle tree. In that case, the nods in the tree will collapse respectively, leading the protocol to treat the modified block as if it were a new one.

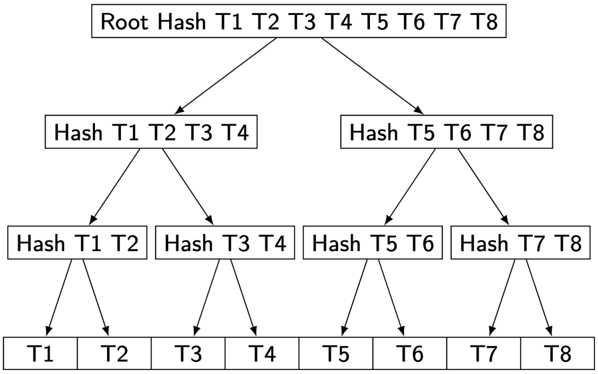

FIGURE 2. Merkle Hash Tree

As shown in Figure 2, a Merkle Hash Tree consists of binary nodes with leaf nodes at the bottom, forming a structure that ensures data accuracy and integrity.

Ethereum needs a tree data structure that can swiftly rebuild a tree root after editing, inserting, or deleting [76], and as such, the tree root must be data-driven not update-driven (regular Merkle trees fail this condition). Ethereum combines a specific trie called a radix trie with a Merkle Tree, known as a Merkle Patricia Trie (MPT). In MPT, there are four kinds of nodes: branch, leaf, extension, and null [77]. A value and a 16-element array make up the branch node. A "branch" is a member of the array that stores a nibble and indexes a corresponding child node. The second node type is the leaf node, which contains a value, byte string, or compressed route called "encodedPath". The third is the extension node, which also has the encoded path and a pointer to the next node. Finally, a null node has an empty string, meaning no contents exist. The full MPT can be condensed into a single cryptographic hash for tamper-proofing (like a Merkle Tree).

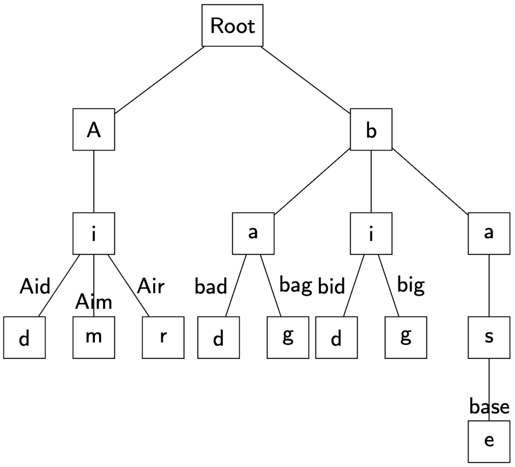

FIGURE 3. Merkle Patricia Trie

As illustrated in Figure 3, Ethereum utilises a Merkle Patricia Trie, combining a radix trie with a Merkle Tree structure to efficiently manage and verify data in a decentralised manner. Furthermore, progress has been made in demonstrating the reliability of data stored on the blockchain. Verkle trees are argued to be a more bandwidth-efficient variant of Merkle trees [78]. As such, a Verkle tree is a Merkle tree variant that uses vector commitments to minimise the proof size of the tree. Concatenating all workflows and signing their cumulative hash could be a different way to ensure the integrity of the total queue.

IV. LAYER 2 SOLUTIONS

The primary approach to achieving blockchain scalability is referred to as *Layer 2*. This approach centres around developing a system that processes transactions off-chain (separately from the main chain and, to a certain level, autonomously), minimising the load on the blockchain and significantly enhancing transaction speeds [79].

Layer 2 solutions have been developed as separate blockchains that inherit security and decentralisation from *Layer 1*, aiming to increase transaction throughput and reduce transaction fees, which *Layer 1* cannot efficiently achieve.

A. ROLLUPS

Rollups are decentralised sidechain solutions created to reduce the load on the main blockchain by executing transactions off-chain in batches and using data compression with smart contracts to scale *Layer 1* chains [80]. Rollups generally store only minimal data on-chain, such as a Merkle root, which helps with on-chain verification and quicker withdrawals. The smart contract keeps the Merkle root (or state root) on-chain, which can be verified using available on-chain data. A new state root is created after each batch of transactions and updates the contract's state if it matches the previous state root. Rollups are generally categorised into two types based on how they prevent fraud and verify the state root: Optimistic Rollups and Zero-Knowledge (zk) Rollups [81].

1) Optimistic Rollups:

Optimistic Rollups work based on the "trust but verify", where they assume the sequencer is acting correctly, and transactions are valid unless they are challenged within a particular period [82]. This approach skips default verification to enhance scalability significantly. However, the contract keeps a record of state root updates and batch hashes [83]. Users can challenge a batch by submitting a fraud-proof on-chain if they identify an error. The contract will reverse the incorrect batch and subsequent batches if the Proof is confirmed.

a: Arbitrum

Arbitrum is an advanced Layer 2 scaling solution for Ethereum that has gained significant interest within the Ethereum community [84]. It operates as an optimistic rollup on the Arbitrum Virtual Machine (AVM), employing multi-round fraud proofs executed off-chain to enhance network performance by efficiently resolving disagreements [85]. This approach provides greater security than alternatives, such as Optimism [86], due to its more sophisticated fraud-proof system. Moreover, Arbitrum supports smart contract development using Ethereum-native tools like Solidity, allowing for a smooth transition of smart contracts. This feature has contributed significantly to the widespread use of Arbitrum for deploying smart contracts.

2) zk Rollups:

The ZK Rollups system employs an off-chain operator to control transactions in batches and produce validity proofs, such as ZKSNARKs, which are then uploaded on-chain to confirm that the state transitions have been executed correctly [87]. Unlike Optimistic Rollups, which assume transactions are valid until proven otherwise, ZK Rollups verify every transaction. Each batch includes a cryptographic proof, called a validity proof, that ensures the new state root matches the outcome of executing the batch of transactions. Although creating these proofs is complex, their verification on-chain is quick. Projects like ZKSync and StarkEX actively explore ZK-Rollups, using SNARK and STARK proofs, respectively [88].

a: ZkSync

ZkSync is a zk-rollup solution successfully implemented on the Ethereum network [89]. As the first ZK-rollup compatible with the Ethereum Virtual Machine (EVM), ZkSync uses SNARK cryptographic validity proofs to enable scalable and cost-effective transactions on Ethereum. While ZkSync may show lower transaction speeds and higher latency, it effectively balances blockchain scalability, security, and decentralisation, achieving significant scalability. Moreover, the inherent zk nature of this solution ensures privacy by default, in this way enhancing the overall security of the system [90].

b: StarkNet

Starknet is a permissionless ZK rollup serving as a *Layer 2* network over the Ethereum blockchain to enhance scalability [91]. This is achieved using ZKSTARKs, a cryptographic technique that eliminates the need for a trusted setup by employing publicly verifiable randomness to create trustless verifiable systems. Furthermore, ZKSTARKs are quantum-resistant and demonstrate greater computational speed and size scalability compared to ZKSNARKs [84].

V. SECURITY AND PRIVACY MODEL OF E-VOTING SYSTEMS

A. TYPES OF VOTING SYSTEMS

Several types of e-voting systems have been developed and used around the world. Some of the most common types are:

a: Direct-Recording Electronic (DRE) voting

DRE voting machines are touchscreen devices that allow voters to make their selections electronically [92]. These machines typically record votes on internal memory and may -or may not- produce a paper trail. DRE machines offer advantages like increased efficiency, faster results, and accessibility features for individuals with disabilities [93]. However, they also pose significant challenges like security vulnerabilities, the potential for tampering, the lack of a verifiable paper trail, and the need for stringent authentication measures.

b: Internet voting

Internet voting has gotten considerable attention as a promising way to overcome geographic barriers and increase voter turnout [94]. Thanks to advancements in digital technologies, it enables individuals to cast their votes remotely using personal devices connected to the internet, such as mobile phones and personal digital assistants [95]. This voting method offers unparalleled convenience, particularly for individuals residing in remote areas or those facing physical disabilities that limit their ability to visit traditional polling stations. Furthermore, internet voting has the potential to streamline the voting process, reduce associated costs, and deliver immediate results. However, widespread Internet voting also concerns the electoral system's security, privacy, and integrity. Its susceptibility to cyber-attacks, the risk of vote manipulation, and the complexity of ensuring voter authentication and anonymity prioritise addressing such issues to ensure the viability and trustworthiness of these systems.

c: Blockchain e-voting Systems

Blockchain technology is an innovation in e-voting, offering enhanced transparency, security, and trustworthiness [96]. By leveraging its decentralised and immutable nature, Blockchain-based voting systems provide decentralised storage of voting records and ensure system robustness and resilience. The transparency of the Blockchain allows for easy review and verification of the voting process, thus reducing the risk of manipulation. The Blockchain's immutability guarantees the voting results' integrity, as transactions cannot be modified or tampered with [97]. Despite challenges related to throughput, privacy, and authentication, Blockchain has the potential to revolutionise elections by providing secure, transparent, and trustworthy solutions for the democratic process.

B. COMMON E-VOTING REQUIREMENTS

E-voting protocols aim to satisfy various security requirements identified in previous studies [98]–[101]. These requirements are essential for developing and evaluating secure e-voting protocols and establishing a reliable voting system that maintains democratic principles.

a: Privacy

Privacy is a fundamental security requirement in e-voting protocols, as it entails that the ballot choices made by voters remain confidential to others, especially the authority overseeing the voting process [102].

b: Integrity

Attempts to tamper with votes must be easily detectable, whether by changing their content or erasing them [103].

c: Authentication

In e-voting protocols, authentication involves verifying a voter's identity through legal identification documents. At the same time, the system must authenticate the identities of election officials to ensure their authorisation to operate the election system [101].

d: Identification

Identification in elections ensures the authenticity of voters and requires individuals to establish their identity as part of the voting process [101].

e: Public verifiability

Universal verifiability indicates that anyone can verify the integrity and accuracy of the final voting result [102].

f: Coercion-Resistance

Making sure voting is free from coercion is crucial for any voting system, as measures should be in place to prohibit any form of coercion during the voting process [101]. Ensuring the integrity of e-voting (e-voting) systems in the face of coercion is of utmost importance.

g: Receipt-Freeness

Voters must be unable to make or acquire a receipt [104]. Vote selling and buying are prevented via receipt-free voting to ensure candidates do not employ voters.

h: Double vote

A registered voter may attempt to vote more than once in a fashion that allows each ballot to be counted [105].

i: Universal verification

Anyone can audit the election and be confident that all votes have been counted and the election has been conducted correctly [106].

j: Scalability

Scalability is vital in e-voting as it can handle expanding data volumes and workloads without compromising performance [107]. It ensures the system can effectively accommodate more concurrent voters as the electorate grows while maintaining optimal functionality.

k: Hardware/software security

Several distinct attack vectors should be considered when designing an e-voting system such as infected voter machines, active network attacks, network correlation attacks, and authorities manipulating votes [108].

l: Robustness

The system should be resilient against failures, attacks, and disruptions, ensuring the voting process can continue uninterrupted [109].

m: Auditability

The system should provide a verifiable audit trail to enable thorough post-election audits and ensure the accuracy of the results [110].

C. SPECIFIC REQUIREMENTS FOR EACH VOTING TYPE

Voting systems must meet specific requirements to ensure the integrity and fairness of the electoral process [102]. In addition to the fundamental requirements of privacy, authenticity, accuracy, security, democracy, and verifiability, different voting types have unique implementation considerations. For DRE-based voting, several requirements related to infrastructure, accessibility, and maintenance must be addressed to ensure a smooth and reliable voting experience.

1) Requirements for DRE Based Voting

a: Affordability

The cost of implementing and maintaining an e-voting system should be reasonable and lower than traditional voting methods to ensure financial viability and encourage widespread adoption [111].

b: Accessibility

DRE voting methods must be accessible to all voters, including those with impairments [112]. Specialised hardware and software should be implemented to accommodate various disabilities, and this allows every individual to participate in the voting process.

c: Maintenance

Regular hardware maintenance is crucial for DRE voting systems, particularly touchscreen devices. With repeated use, touch screens can wear out, necessitating proactive maintenance and replacement to ensure the reliability of the voting process.

2) Requirements for Internet Voting

a: Mobility

Internet voting offers flexibility and convenience and allows voters to cast their ballots from anywhere with an internet connection [113]. This inherent mobility feature empowers individuals to participate in the electoral process more actively despite geographical constraints. By leveraging the convenience of internet voting, citizens can exercise their voting rights and contribute to the democratic process without being bound by traditional limitations of in-person voting methods.

b: Flexibility

Internet voting systems should accommodate various devices and network connections. Voters should be free to use different devices, such as desktops, laptops, palmtops, mobile phones, and different network types, including Ethernet, dial-up connections, and wireless networks [113].

c: Transparency

Internet voting systems should enable independent observers to verify the validity of each vote without revealing the candidate to whom the vote was cast. This transparency ensures the integrity and credibility of the electoral process [114].

d: Advocating for the Implementation of Various Voting Client Software Solutions

Acknowledging that certain voter demographics may have concerns regarding the dependability of officially provided voting client software, as well as apprehensions about their proficiency in using the software, the availability of alternative software solutions becomes advantageous in both situations [115].

VI. EXISTED E-VOTING SYSTEMS

This section explores e-voting and focuses on three main types: DRE systems, Internet E-voting, and Blockchain E-voting. These systems include key voting stages like voter setup, casting ballots, and tallying votes. Our goal is to understand the complexities of e-voting comprehensively. We will also evaluate relevant studies for each system and provide insights into their pros and cons. By analysing these standard types and assessing research merits, we aim to provide an in-depth understanding of e-voting. To simplify this understanding, we have included three tables throughout the paper. Table 1 outlines the fundamental workings of each electronic voting system. Table 2 examines the security and privacy features of these systems, with a focus on factors such as voter privacy, verifiability, prevention of double voting, and scalability. Table 3 provides a comparative assessment of the systems based on these criteria, highlighting their differences and similarities. Together, these tables offer a comprehensive overview of the critical aspects of electronic voting systems.

A. DRE

1) Every Vote Counts

This paper introduces DRE-i, a novel TA-free End-to-End (E2E) verifiable e-voting protocol called DRE with Integrity, leveraging encryption techniques for tally verification [116]. It ensures integrity by publishing additional audit data and offers a fail-safe mechanism for recovery from missing or corrupted ballots.

Technical Specifications:

- **Setup:** A secure e-voting system uses parameters (p, q, g) within a multiplicative cyclic group. Each DRE machine generates a private signing key stored in a tamper-resistant module, certified by a trusted entity, to prevent fake votes. The system creates a tabular structure of encrypted ballots, ensuring confidentiality, integrity, authenticity, and non-repudiation. Unique public keys are generated for each ballot, with cryptogrammes representing “Yes” and “No” votes, accompanied by ZKP for security and privacy. The system is optimised for efficiency and memory usage, ensuring a smooth and swift voting process.
- **Voting Process:** Authenticated voters use random tokens to interact with DRE machines. The voting process consists of two steps:
  1) Selection and Commitment Printing: The voter selects their choice on the touch-screen. The machine prints a commitment containing the ballot’s serial number and the encrypted choice (cryptogram), digitally signed for authentication.
  2) Confirmation or Cancellation: The voter can confirm or cancel their vote. Cancellation allows viewing the selection in plain text and casting a “dummy” vote, ensuring accurate recording and detecting potential machine cheating. Voters receive a receipt with the ballot serial number and encrypted vote, which can be compared to entries on a public bulletin board for verification. Unused ballots are marked as “dummy” votes, supporting voter-initiated challenges and system verifiability.

TABLE 1: Comparison of Frameworks Used DRE, Internet and Blockchain Approaches

#	Framework	Approach	Research Method	Tallying	Evaluation	
					Advantages	**Drawbacks**
1	[116]	DRE	Encrypted voting with public keys; end-to-end verifiability, homomorphic tallying.	self-tallying	Elimination of tallying authorities; End-to-end verifiability; Enhanced security; Improved efficiency; Better usability.	Challenge of pre-computation technique; Increased cost due to tamper-resistant hardware; Difficulty in creating a secure API; Privacy risk from breach of secure storage module.
2	[117]	DRE	Ranked voting, private DRE machine, Borda count, voter privacy.	self-tallying	Reflects Voter Preferences; Accommodates Diverse Opinions; Maintains Voter Privacy; Limits Compromise Impact; Addresses Coercion Attacks; Provides End-to-End Verifiability; Enhances Electoral Robustness.	Centralised Setup; Device Knowledge of Voter Preference.
3	[118]	DRE	Generate voting cards, verify IDs, employ RSA encryption, tally with private key decryption under supervision.	3rd-party	Scalability; computational efficiency; flexibility in device selection; constant complexity cryptographic operations; efficient tallying; usability on resource-constrained devices; consideration of organisational aspects for scalability.	Lack of discussion on potential limitations or drawbacks; inadequate comprehensive evaluation of potential challenges or shortcomings.
4	[119]	Internet voting	Secure e-voting with key-based voter IDs, encrypted ballots, proofs, and verifiable tallying for integrity and transparency.	self-tallying	Simplicity; Vote verification; Minimal impact on client-side operations; Successful implementation as a new voting platform.	Lack of end-to-end verifiability; Dependency on voting server for accuracy; Potential for fraudulent vote additions.
5	[120]	Internet voting	Setup distributes private keys, voting involves blinded encryption, and tallying employs smartphone verification and DS public key decryption.	3rd-party	Improved reliability, verifiability, and robustness of the Norwegian protocol modifications; Enhanced protocol performance maintaining vote privacy despite potential corruption or cooperation between BB and RG.	Lack of complete anonymity guarantees, permitting privacy breaches by linking pre- and post-receipt codes; Need for future advancements to address the significant issue of privacy in voting protocols.
6	[121]	Internet voting	ElGamal key generation, NIZKP verification, aggregated public keys for encryption, multi-factor authentication, encryption for secure vote transmission and cooperative decryption	self-tallying	Secure, verifiable, and valuable e-polling system; Demonstrated performance, security, and comparative attribute analysis; Integration of individual verifiability through voting receipts.	Increased complexity from added key management and storage overhead in the double verification process; Vulnerability to voter coercion attack, potentially influencing voter's selections through voting receipt exploitation.
7	[122]	Blockchain	Public key-based key generation, blinded certificate issuance, CA-signed eligibility verification, new key pair generation, multi-entity vote signing, involving inspectors.	3rd-party	Enhanced security; Transparency; Scalability; Flexibility; Data integrity.	Potential IP address exposure; Lack of receipt-freeness; Vulnerability with a single inspector; Compromised inspector's private key; Single point of failure; Susceptibility to attacks.
8	[123]	Blockchain	Paillier encryption scheme; Secret sharing techniques; Decryption and signature validation for tallying	3rd-party	Meets key e-voting requirements; Enables active participation; Enhances confidence through smart contract transparency.	Lack of specific standards; Unclear handling of voter coercion, vote buying, double voting, and privacy concerns; Potential need for improved effectiveness and trustworthiness

Continued on the next page

TABLE 1 – continued from previous page

9	[124]	Blockchain	collaborative parameter generation, cryptography (ring signatures), and verification protocols to ensure privacy, anonymity, and integrity in voting	self-tallying	Receipt-freeness; Privacy enhancement; Coercion resistance.	Cryptographic complexity; Verification issues; Scalability challenges.
10	[125]	Blockchain	Shamir's secret sharing scheme; Circle Shuffle technique; P2SH; Multi-signature scripts; Voting Commitment Transaction (VCT); Recover Transaction (RT).	self-tallying	Receipt-freeness challenge tackled; Privacy enhancement through masking identities and thwarting linkages; Coercion resistance for secret balloting.	Cryptographic complexity affecting efficiency and resources; Verification issues compromising integrity; Scalability challenges demanding optimisation.
11	[126]	Blockchain	ZKP, blockchain technology (Ethereum), and mathematical computations (discrete logarithm)	self-tallying	Easy Setup; Cost-effectiveness ($0.73 per voter); Maximum Privacy; Public Verifiability; Correct Execution Assurance; Reliability Enhancement.	Blockchain Network Fees; Limited Scalability (50-60 voters).
12	[127]	Blockchain	certificates for auditing, code-based cryptography for security, and traceable signatures for anonymity.	self-tallying	Voter Anonymity; Auditability; Fairness and Correctness	Limited Scalability; Precision; Rigid for Voter Diversity; Complexity; Quantum Resilience
13	[128]	Blockchain	ZKP, privacy-preserving methods, P2P network, IoT devices, gateways, and supplier participation.		The system offers transparency in IoT software update voting processes; Fairness.	Can be expensive to implement, particularly when numerous IoT devices are needed to support the system.
14	[129]	Blockchain	ElGamal Cryptosystem, Neff Shuffling Method, ZKP	self-tallying	Technologically advanced platform; User-friendly interface; Improved election security; Enhanced transparency; Increased efficiency; Advancement of democratic principles.	Lack of precision with multiple options; Vulnerability to coercive tactics; Security risks from external observers.
15	[130]	Blockchain	Token Randomiser, zkSNARK Proof		Coercion Resistance; Efficient Tallying: Streamlined vote counting	Limited Real-World Testing; Trust in Authorities; Cost Implications; Legal Compliance; Usability for Diverse Demographics.
16	[131]	Blockchain	Integration of SoftHSM, Identity Mixer Transition	self-tallying	The network structure is flexible, adapting to various scenarios. Security is bolstered by SoftHSM, and privacy is preserved. Transition plans to Identity Mixer offer improved privacy.	Complexity can hinder deployment, increase costs, and affect privacy. User education is needed for adoption, and cybersecurity remains crucial.
17	[132]	Blockchain	NIZKP, re-encrypting them using a Randomiser (R), Cryptographic Shuffles	self-tallying	Flexible Vote Encoding; Reduced Computational Burden	Complexity and Runtime; Lack of Coercion Resistance
18	[133]	Blockchain	Ring Signatures, Random Sortition, Threshold Cryptography, Rewards and Penalties:	self-tallying	Privacy-Preserving; Verifiability; Robustness; Incentive Mechanism.	Complexity; Scalability Concerns; User Adoption; Regulatory Compliance.
19	[134]	Blockchain	Client API, Elliptic Curve, Transaction Batching, NIZK, Pseudo-Random Assignment	self-tallying	Scalability; Privacy; Security; Verifiability.	Handling unanimous votes presents challenges while protecting privacy; Potential deanonymisation risks through IP address linkage.

- **Tallying and Results Verification:** The system multiplies all published ciphertexts for dummy votes, resulting in $g^{\sum_i v_i}$. A key feature is the cancellation technique, which combines $\sum x_i y_i = 0$ with homomorphic encryption. This self-enforcing protocol eliminates the need for secret keys. The "yes" vote count ($\sum_i v_i$) is publicly verifiable by comparing g^{β} and $g^{\sum_i v_i}$. Dummy votes (λ) are also verifiable. "No" votes are calculated as $\alpha = n - \beta - \lambda$, where n is the total number of ballots.

Security Analysis:

- **Strengths:** This technique offers several notable advantages. The need for tally authorities is eliminated by employing encryption to secure votes. After the election, multiplying the encrypted votes removes random components, allowing anyone to verify the accuracy of the final count quickly. Voters receive receipts to cross-check with a public bulletin board, supporting end-to-end verifiability. Implementing multiple security measures, such as encryption, mathematical formulas, and proofs, enhances the system's integrity and resilience against potential threats. Furthermore, aligning the voting process with traditional methods improves usability by reducing complexity.
- **Weaknesses:**
  The pre-calculation method requires secure data storage and access throughout the voting period [135]. Tamper-resistant hardware is essential to safeguard critical election processes, and adopting such hardware can significantly increase the costs of each DRE device. Moreover, developing a robust API for this hardware presents considerable challenges. Any compromise of the secure storage module can compromise the confidentiality of all ballots, posing a significant risk to the system's security.

2) E2E Verifiable Borda Count Voting

The proposed DRE system [117] achieves end-to-end (E2E) verifiability for Borda count elections without relying on tallying authorities. It minimises information leakage by revealing only the total score of each candidate and mitigates Italian attacks. Even in the presence of compromised DRE machines, the integrity of the tallying result remains intact and limits the adversary's knowledge to the partial tally at the time of compromise.

Technical Specifications:

- **Setup:** In order to cast a vote, A voter v_i has to authenticate with the polling station. Each voter will get a random password to log in to their account and vote in a private room. The DRE machine has two $1 \times n$ vectors $S = (s_1, \cdots, s_c)$ and $U = (u_1, \cdots, u_c)$, both initialised to zero $(0, 0, \cdots, 0)$ where c representing the candidates who are contesting the election.
- **Voting Phase:** A voter ranks the candidates using the interactive interface provided by the DRE machine. For each rank, a fixed score is assigned. Several scores are indicated as $a_1, a_2, \cdots, a_c$ depending on c where c denotes the number of candidates in the election. The first candidate's score will be assigned as c, the second candidate's score is c_1 and so on. The set of votes from the voter is denoted as $V_i = (v_1, v_2, \ldots, v_c)$. After the voter keys in his choice V_i, where $i \in [n]$, the machine will choose random numbers X_i to compute $\langle B_i, X_i' \rangle$. This $\langle B_i, X_i' \rangle$ will be part of the digitally signed voter receipt. It will be printed on paper, and the voter needs to audit his ballot or confirm it to the machine. If you choose to audit the ballot, the second part of the receipt X_i and V_i will be printed and covered by a digital signature. The ballot will be marked as audited, and the receipt will be posted to the bulletin board marked as an audited ballot. If the voter checks that the revealed vote V_i is not the same as his choice, The voting process then restarts from the beginning, allowing the voter to make a new decision. If the voter confirms his vote, NIZK proof will be generated and printed on the receipt with the digital signature. The NIZK proves that$(v_{i1}, v_{i2} \cdots .v_{ic})$is a permutation of $r = (a_1, a_2, \cdots, a_c)$. S and U will be updated as S=$(s_1, s_2, \cdots, s_c)$, U=$(u_1, u_2, \cdots u_c)$, where U show how many votes each contender received overall.
- **Tallying phase:** Following the election, the DRE machine posts S and U on the notice board. The election's winner can be selected from $u_j = max(u_1, \cdots, u_c)$, where u_j indicates the total number of votes cast for the candidate j for all $j \in [1, c]$. The method ensures the privacy of votes and the accuracy of the tally. An attacker can only discover partial counting at the moment of intrusion if a DRE machine is completely hacked.

Security Analysis:

- **Strengths:** The DRE-Borda system has several benefits. These include accurately capturing what voters want, representing many views, keeping voters private, reducing compromise influence, fighting off coercion attacks, and allowing full checkability through the voting process. The primary objective of these entities is to ensure the preservation of the election's integrity by checking the congruence between the recorded outcomes and the genuine votes. Furthermore, they attempt to maintain some unpredictability and ensure the final count's accuracy. In protecting the confidentiality of voting records, the intruder's privileges are limited only to partial tabulations until the moment of penetration in a comprehensive breach of a DRE machine; integrating these characteristics enhances the election procedure, thus increasing its durability and inclusivity.
- **Weaknesses:**It is essential to recognise that the efficacy of these systems depends on a centralised configuration in which a central facility physically captures and records votes [136]. In this particular situation, it becomes inevitable for the touch-screen device to gather information regarding the voter's preferences.

3) SecureBallot

The authors in [118] introduce SecureBallot as a secure open-source e-voting system that disconnects voter ID from voting.

Technical Specifications:

- **Setup:** Staff produce and configure voting cards by listing candidates and the maximum number of choices. Electoral officers oversee ID cards, identify voters, check if they are eligible and have not already voted, offer voters an explanation and an unlocking token, and ensure the vote went smoothly.
- **Voting:** The voter can access the designated voting booth, gain entry to the voting station by employing the unlocking token, and select their preferred candidates after obtaining authorisation from an election official. To ensure the confidentiality and integrity of each vote, a symmetric key is generated randomly for every voting packet. Subsequently, the random key is encrypted using the RSA algorithm while employing the public key specific to the ongoing election. To fulfil the requirements for security, integrity, and consistency, the HMAC algorithm is utilised using SHA-256 to generate a digest of the voting package and encryption key. Following this, the digests are securely transmitted to the central voting machine in an anonymous manner.
- **Tallying:** During the initial tallying stage, the private key specific to the ongoing election is employed to decrypt each symmetric key, thereby revealing the contents of the corresponding voting cards. The vote-counting process occurs at this stage after determining the election results. The recorded outcomes are stored in XML format and subsequently subjected to digital signing by the Staff members. It is important for the notary possessing the private key associated with the election to be present to conclude the tallying phase successfully. Figure **??** shows the operations involved in the SecureBallot voting phase.

Security Analysis:

- **Strengths:** SecureBallot offers several advantages, including scalability, computational efficiency, flexibility in device selection, constant complexity cryptographic operations, efficient tallying, usability on resource-constrained devices, and consideration of organisational aspects for optimal scalability. These features contribute to making SecureBallot a robust and accessible e-voting system suitable for elections of various sizes.
- **Weaknesses:** While steps have been taken to close a regulatory gap that allowed companies to misuse sensitive user data in e-voting systems, more needs to be done. A new approach is necessary to collect important information while protecting users' privacy [137].

B. INTERNET E-VOTING SYSTEM

1) Neuchâtel's Scheme

In an election with a single vote casting, [119] outlined a straightforward ballot-casting protocol that uses return codes to allow voters to check their ballots.

Technical Specifications:

- **Setup:** The election administrator creates unique voter identities for the election by generating specific keys and publishing them on the bulletin board. Private keys are distributed to authorised individuals, while voters provide their identities to the registrars for verification. Upon successful verification, voters receive their key pairs, encrypted values, and cryptographic proofs. The bulletin board displays public keys and proofs for transparency and verification purposes.
- **Voting Phase:** The voter authenticates and selects their choices using a voting device. The ballot is generand voter identification areed and sent to the server along w ballot is successfully processed, the ballot box is updated. Return codes are generated and used to update the ballot state. The voter confirms their vote, and a finalisation code is generated and stored with the ballot. The server updates the ballot state and sends the finalisation code to the voter's device for verification. If the code matches, the submission is confirmed correct.
- **Tallying phase:** The election administrator runs the Tally interactive protocol and verifies the proof of correctness using the parameter π. If the output is 1, the outcome is declared fair. If not, an investigation is launched to determine the cause of the failure.

Security Analysis:

- **Strengths:** The proposed ballot-casting protocol provides a simple and user-friendly process for voters while ensuring vote verification by allowing voters to compare the finalisation code supplied with the value obtained during registration, enhancing the security of the voting process. The client-side procedures are optimised by redistributing one of the server-side functions to the voting client programme, effectively reducing the number of processes required for the application and improving efficiency. The successful implementation of this unique voting platform is a significant achievement in the field of election technology, demonstrating its potential for real-world applications.
- **Weaknesses:** The protocol lacks complete end-to-end verifiability, requiring voters to trust the voting server for accurate vote counting and fraud prevention. This reliance on a centralised server introduces manipulation risks and reduces the transparency of the election process, as voters and auditors need help to verify the results independently. Additionally, while simple, using return codes requires help with complex ballots and incurs high costs for secure printing and delivery [138]. The system's dependence on trusted printing and delivery services creates a significant vulnerability, as leaked verification codes could allow vote tampering.

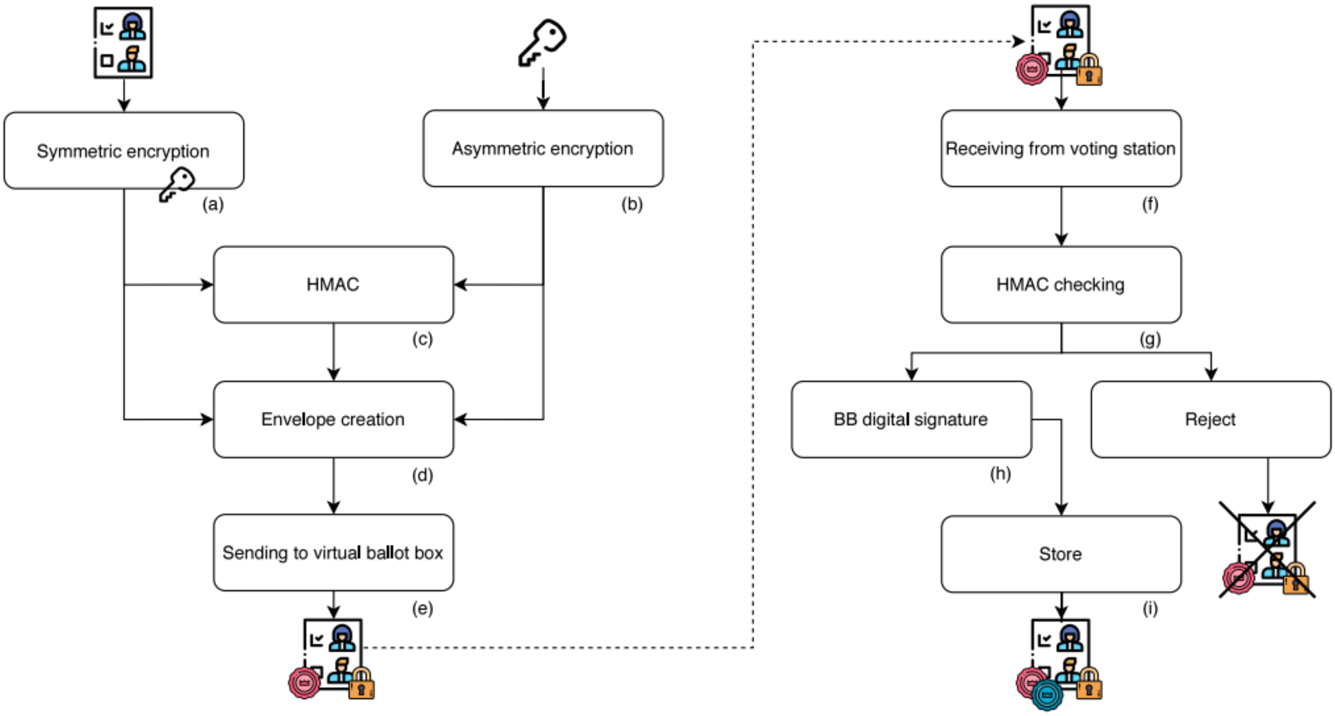

FIGURE 4. ***SecureBallot Voting phase operations [118].***

2) Norwegian internet voting protocol revisited

In 2016, [120] suggested a way to increase the accuracy and privacy of the Norwegian system [139] where they propose the following method to prevent the issue of obtaining the private key of the Decryption Service if any cooperation occurs between the former participants, Ballot Box (BB) and Receipt Generator (RG) (without including additional players).

Technical Specifications:

- **Setup:** The proposed voting method is very different from Norway's in that the vote against BB and RG is done secretly using their method. Following this, RG distributes the private key pairs c_1 and c_2, and the public key is represented by $h_1 = g_1^{c_1+c_2}$. A unique key pair $(\alpha, h_2 = g_2^{\alpha})$ for a homomorphic cryptosystem will be stored on the decryption server (DS).
- **Voting:** In order to blind a vote, PC selects a random value $(f(v_j)^{\ell_i} \forall_j = 1, \cdots, k)$. The blinded votes are then encrypted using the public keys of BB and RG. Then, using the DS public key, the PC calculates the second encryption of the same votes without blinding. The post receipt code is sent to the voter via SMS on the voter's smartphone once BB and RG decrypt the first encryption. It should be noted that casting blind votes alters the outcome of the receipts (r_j'), and RG would produce and send r_j^{ℓ} instead of r_j'.
- **Tallying:** Once the pre-receipt code $v_j, f(v_j)^{s_i}$ and post-receipt code $(r_j'^{\ell_i})$ have been received. In order to complete the verification process, the smartphone uses a 2D barcode scanner (such as a QR code) to read the voter's selected receipt codes $(f(v_j))$ as well as the ℓ_i that is displayed on the screen of the PC. To compare the findings with the post-receipt code obtained by SMS, the application will compute $(f(v_j)^{s_i})^{\ell_i}$. If the voter completes the verification process via a smartphone, the second encryption, which employs the DS public key, is sent for decryption and tallying.

Security Analysis:

- **Strengths:** The study presents a comprehensive set of improvements to the Norwegian protocol, enhancing its dependability, verifiability, and overall resilience. These modifications address and eliminate underlying assumptions within the protocol's architecture. The findings demonstrate significant enhancements in the procedure's effectiveness, ensuring the highest level of vote confidentiality, particularly in scenarios where there is a potential for corruption and collusion between the BB and RG.
- **Weaknesses:** The distribution of receipt codes using both a pre-channel, such as a postal service, and a post-channel, employing SMS, raises concerns regarding the practical security of the procedure. Using these two transmission modes introduces potential vulnerabilities that could compromise the integrity of the voting process. Furthermore, it is crucial to recognise that the equation $a_3 = a_1 + a_2$ does not adequately address the fundamental issue of trust, which constitutes a significant weakness within this specific context. The reliance on this equation leaves the system susceptible to potential

breaches of trust, undermining the overall security and reliability of the protocol.

3) SeVEP

[121] introduced an innovative e-voting system that leverages zero watermarking and encryption technologies to establish the integrity of both votes and voter identification.

Technical Specifications:

- **Setup:** In the initial setup stage, the polling organisation (PO) generates the requisite public and secret keys utilising the ElGamal cryptosystem. Each participating PO generates its unique portion of the cryptographic key and uploads the corresponding public component, accompanied by NIZKP, onto the BB. Subsequently, the BB verifies these proofs and aggregates the individual shares to form a consolidated public election key (Kppo). This aggregate key allows voters to securely encrypt their ballots before submission during the election process, thus ensuring integrity and confidentiality within the voting system.
- **Voting:** The cooperation of all POs involved protects the encrypted votes. A specific key, K_sPO, is required for decrypting votes encrypted with the public election key, K_pPO, which can only be used when all POs collaborate. To participate in the voting, voters must register and obtain a certificate proving their identity. This process involves a multi-factor authentication system to ensure secure access. The ballots provided to voters are devoid of voting labels or watermarks, which helped prevent the identification of individual voters based on altered votes. Within the indicated voting period, voters have the option to cast multiple ballots to a maximum of three. However, only the results of the last ballot cast by each voter will be considered valid to prevent instances of double voting. During the polling phase, confirmed votes are posted by the Polling server on the BB. These posted votes are kept confidential, safeguarding the privacy of the voters, and maintaining the integrity of the election process.
- **Polling phase:** In the mixing and tallying phase, the POs sequentially shuffle and re-encrypt the input list to prove the accuracy of their actions to the BB. The BB then verifies these proofs and shares the shuffled list with the next PO, repeating this process across all POs. Once all proofs are successfully validated, each PO decrypts their share and creates a plaintext ballot list (BVk) -they individually prove the correctness of decryption to the BB-. The primary PO then performs factorisation to determine voting options, and these options are published on the BB along with associated 3-digit numbers (γ_{Vk}) for each polling question.

Security Analysis:

- **Strengths:** Their system is quite secure, verifiable, and valuable e-polling as demonstrated by its performance, security, and comparative examination of security attributes and cryptographic costs.
- **Weaknesses:** Adding key management and storage overhead to the double verification process results in even more complex computations [140], [141]. Moreover, within the context of the threat model (which is an identified potential attack is voter coercion) whereby an individual with coercive intent may manipulate and influence a voter's selection of voting options, the success of this attack is contingent upon the voter possessing a voting receipt, which the polling organisers typically provide to enable individual verifiability.

C. BLOCKCHAIN BASED E-VOTING SYSTEMS

Many researchers have discussed the feasibility -and the requirements- of an End-to-end voting system based on Blockchain Technology for real-world elections.

1) An E-voting Protocol Based on Blockchain

[122] propose a decentralised e-voting protocol based on Blockchain technology that eliminates the need for a trusted third party.

Technical Specifications:

- **Setup:** In the initial stage of the election process, every voter generates a pair of public and private keys. Voters must submit their public keys and necessary personal information to the Certificate Authority (CA) for authentication purposes. Once all registered voters are verified, the organisers publish a list of eligible voters' public keys. Each voter receives two blinded certificates containing public keys, signed by the organisers and the Inspectors. Voters prepare a cryptographically secure voting message, the "vote string" V, consisting of the "Choice Code" C, a series of zeros $0000\ldots0000$ as padding, and a 'Random String' R. This is mathematically formulated as:

  $$V = C \,||\, 0000\ldots0000 \,||\, R \tag{1}$$

  The vote string is hashed using a cryptographic hash function, like SHA-256, to produce $H(V)$.

- **Voting:** During the voting stage, the CA verifies each voter's eligibility and checks for previous ballot casting. Eligible voters encrypt their hashed vote string $H(V)$ with their private key k_{private}, creating a digital signature σ:

  $$\sigma = E_{k_{\text{private}}}(H(V)) \tag{2}$$

  This signature and the new public key are transmitted over the blockchain network. Each vote is included in a block and validated via a consensus algorithm like PoW or PoS. The vote is further signed by relevant entities and sent to inspectors for their signatures.

- **Tallying and Results Verification:**
  After voting, the CA decrypts and validates the votes.

Each vote's digital signature σ is verified using the voter's public key k_{public}:

$$D_{k_{\text{public}}}(\sigma) = H(V) \tag{3}$$

Votes are tallied as per the election rules, with the final tally represented by:

$$T = \sum_{i=1}^{n} V_i \tag{4}$$

where T is the total number of votes, n is the number of valid votes, and V_i each valid vote. The CA then announces the election results.

Security Analysis:

- **Strengths:** The blockchain-based e-voting system eliminates the need for a trusted third party. It involves an organiser and an inspector, with the inspector ensuring the organiser's integrity. Blockchain enhances security, flexibility, data integrity, and transparency, addressing common issues in traditional voting systems.
- **Weaknesses:**
  The system may expose voters' IP addresses, compromising vote confidentiality and receipt fairness. Additionally, it allows for the observation of the total number of votes before actually casting a ballot, which has the potential to lead to voter manipulation [142]. Relying on a single inspector creates a potential single point of failure, as compromising the inspector's private key could undermine the entire election process. Despite these limitations, the protocol generally meets most imposed requirements.

2) decentralised E-Voting Systems

A decentralised e-voting system built on the Blockchain was presented in [123]. **Technical Specifications:**

- **Setup:** Each user picks a random number t, generates unique user code PID_i and transmits it to Recording Centre (RC) for verification. RS issue a voting certificate $Cert(V_i)$= $PID_i, Sig_{d'}(PID_i)$ to V_i and publish the PID_i of eligible voters onto the bulletin board. The bulletin board is implemented by the smart contract (SC).
- **Voting Phase:** During the voting phase, each voter will obtain a ballot signature and personal key pair from Authentication Server(AS). AS generates V_i''s Paillier-based public/private key pair (pkv_i, skv_i) and sends it back to V_i. V_i encrypts his vote and sends it to AS.
- **Encryption and Storage Phase:** Data will be encrypted by Voting Website (VWeb) using V_i's public key pk_{V_i} and stored in Distributed Data Servers (DDS) with PID_i. Following voting, the candidate number (λ) will be split into k plaintext coordinates through $(3,5)$ secret sharing technique that can be deduced from 3-out-of-5 plaintext coordinates. After receiving the coordinates, DDS will encrypt the coordinates using RC public key pk_{RC} then announce the coordinates and PID_i to the SC for V_i to verify the correctness of the tallying.
- **Tallying Phase:** SC decrypts the votes during the tallying stage by using sk_{V_i}. Next, it extracts λ using the $(3,5)$ secret sharing scheme to confirm the validity of the ballot signatures.

Security Analysis:

- **Strengths:** The work presents secure evoting systems that meet key requirements. It enables active participation and fosters confidence through smart contract transparency.
- **Weaknesses:**The main flaw of this scheme is its lack of fairness [143]. Furthermore, the system's scalability is negatively impacted when users increase, resulting in increased overhead [144]. The design lacks clear standards and detailed provisions concerning voter coercion, vote buying, double voting, and privacy. Resolving these issues would significantly improve the system's effectiveness and enhance its trustworthiness in advancing democratic practices.

3) Hyperledger Fabric Permissioned Blockchain

The authors in [124] present a conceptual framework combining e-voting and Hyperledger Fabric Permissioned Blockchain.

Technical Specifications:

- **Setup:** The organisation's administrator defines voting questions and eligible departments that can participate in the voting process. This information is then securely saved on the ledger. Following this, authorities from the department gather essential details related to the voting process, including specifics on the voting time and lists of eligible voters, and afterwards record this data in the ledger. In addition, a pair of key values (public and private keys) are generated for each voter. The public key is then recorded on the ledger, while the private key is securely stored within a private data collection in Hyperledger Fabric, which has limited accessibility. This methodology establishes the foundation for a reliable and accountable voting procedure within a blockchain environment.
- **User Registration:** The registration process begins with users generating a cryptographic key pair and a random number. Users then send an encrypted combination of the random value R and a hash of their public E_v and private D_v keys to the registration logic component. This data is cryptographically validated using the department's private key $D_{dep,i}$. The result is a blind signature, a key aspect of the process, which serves as proof of registration. The relevant information is securely recorded on the ledger. Users then anonymously submit their signed public key$D_{dep,i}(h(E_v))$ along with their public key (E_v) to the registration component. This method ensures user privacy while verifying registration legitimacy and commitment to the electoral process.
  The system also allows for registration revocation. Users can opt out of digital voting in favour of traditional

paper methods by formally requesting the registration component to mark their registration as “revoked”. This approach preserves the registration record, enabling potential re-registration while removing the user from the digital voter list and allowing participation via conventional paper voting.

- **Voting and Tallying:** In the Voting phase, individuals who have completed the registration procedure can securely transmit encrypted ballots to a chain code. These individuals also have the option to modify their selection before the conclusion of the voting process. During the second step of Result Counting, participants possessing the required private key components are responsible for uploading them to the chain code. This action facilitates the decryption of ballots and eventually allows for the publication of voting results on the ledger. The implementation of this measure guarantees the preservation of the voting process' integrity and transparency.

Security Analysis:

- **Strengths:** The system efficiently uses existing certificates for voter eligibility, enhancing security through restricted key possession and departmental signatures. Its high verifiability is a key advantage, with each vote recorded in a ledger using unique user IDs, allowing voters to confirm their votes' inclusion independently. This transparency significantly bolsters the voting process's integrity and trustworthiness.
- **Weaknesses:** The two-phase registration process may exclude voters who do not complete both stages from all voting methods. While secure, this approach is inflexible. Additionally, there are concerns surrounding privacy and security. For transactions to be completed, both parties must either be online simultaneously or lock their tokens, potentially creating further complications [145].

4) SHARVOT

To ensure transparency and privacy in the voting process, [125] presented a novel e-voting system using Blockchain and secret sharing techniques.

Technical Specifications:

- **Setup:** The SHARVOT protocol employs a combination of Shamir's secret sharing scheme and the Circle Shuffle technique to maintain voters' privacy. To ensure this, the protocol assigns private key shares to voters through a t-of-n threshold scheme, with the dealer acting as the assigning authority. These key shares are distributed to voters who have committed fees in an n inputs - 1 output transaction, which serves as a means of storing and permanently recording the votes. Integrating Shamir's secret sharing scheme and the Circle Shuffle technique within the SHARVOT protocol aims to enhance the confidentiality and integrity of the voting process.
- **Voting:** The voting phase commences with assigning a unique public/private key pair to each candidate, denoted as A and B, respectively. Simultaneously, the dealer publicly discloses the public keys P_A and P_B while maintaining the confidentiality of the corresponding private keys k_A and k_B. Subsequently, the dealer applies a computation to generate a total of n key shares for each secret and proceeds to distribute a pair of key shares, denoted as $(k_{A,i}, k_{B,i})$, to each voter U_i. Each voter, represented as U_i, employs encryption techniques to encrypt their ballot, indicating their preferred candidate, and forwards the resulting list of shuffled votes to the dealer.
- **Tallying:** In the tallying phase, the dealer generates a P2SH address using selected key shares from voters. The P2SH script incorporates if-else statements, multi-signature scripts for up to 13 votes and candidate public keys. A Voting Commitment Transaction (VCT) is created, containing an output of $n \times x$ Bitcoins sent to the P2SH address. Voters add their input and sign the VCT. If a candidate decrypts $t + 1$ or more key shares, they can access Bitcoins. If no candidate obtains enough key shares, voters can broadcast a Recover Transaction (RT) to retrieve their fees. The protocol ensures the recoverability of committed Bitcoins if candidates lack sufficient key shares to spend the transaction.

Security Analysis:

- **Strengths:** The SHARVOT protocol utilises Blockchain technology to address voter privacy, eligibility, and ballot integrity matters. Implementing transaction shuffling and encryption techniques achieves voter anonymity and a permanent record on the Blockchain. The dealer certifies eligibility through key distribution, and encryption protects against manipulation.
- **Weaknesses:** The system faces the issue of multiple vote submissions, where a participant could submit duplicate or fraudulent votes to disrupt the voting process [146]. Despite its innovative approach, the protocol relies on a central authority and poses a security risk if compromised. These limitations emphasise the need for further improvements to enhance security and effectiveness in the voting process.

5) A Smart Contract for Boardroom Voting

A ZKP E-voting protocol based on Ethereum was proposed by [126] and used in small-scale voting scenarios.

Technical Specifications:

- **Setup:** The election administrator, representing the smart contract owner, verifies voters through user-controlled accounts and updates the eligible voter list. Timers are enforced to ensure timely progress. Registration requires eligible voters to deposit ether and provide key commitments. Administrator-defined timers include:
  1) **finishRegistration**: Voter key registration deadline.
  2) **beginElection**: Start of the election.

3) **finishCommit**: Deadline for vote commitments (optional).
4) **finishVote**: Deadline for casting votes.
5) π: Minimum time for commitment and voting stages.

- **Voting:** In the voting process, eligible voters register using their voting keys and ZKP while also submitting a deposit. The administrator initiates the transition from the SIGNUP stage to either COMMIT or VOTE. Ethereum computes reconstructed voter keys during this transition. For the COMMIT stage, voters hash their votes and publish commitments on the Ethereum blockchain, leading to the contract's advancement to the VOTE stage upon final commitment acceptance. During this stage, voters publish encrypted votes alongside ZKP and refunds are granted for accepted votes. Ethereum performs tally computation upon receipt of the final vote and finishes in determining the election outcome.
- **Tallying:** The administrator triggers Ethereum to compute the tally by multiplying all votes and finding the discrete logarithm to count yes votes. Deposits encourage voter participation and refunds for compliance based on timestamps. Each stage of the election is described in Figure 5:

Security Analysis:

- **Strengths:** The paper introduces a solution that provides a simple setup, cost-effectiveness (about $0.73 per voter), robust privacy protection, and can only be compromised by full collusion among all other voters. Also, it allows public checking using a public bulletin board to ensure everyone in the voting process sees the same information. Significantly, this implementation represents the first decentralised internet voting protocol on the Ethereum Blockchain for accurate execution and increasing trustworthiness.
- **Weaknesses:** The suggested approach cannot prevent fraudulent miners, compromising the system's integrity [147]. A malicious voter could also bypass the voting process by submitting an invalid vote. The protocol offers no guarantees for resistance to coercion, leaving trust in the hands of the electoral administrator. Furthermore, the voting process may incur additional expenses due to Ethereum network fees. Scalability is also a concern, as the protocol is limited to handling a small number of voters, constrained by the Ethereum platform's inherent limitations and the substantial computational cost of homomorphic encryption [148].

6) Anti-Quantum E-Voting Protocol

A Blockchain-based e-voting protocol that allows for transparency and visibility in the voting process was proposed by [127].

Technical Specifications:

- **Setup:** the voting content and candidate roster are publicly disseminated on the blockchain, ensuring transparency. Eligible voters then register by providing their ID and public keys to the regulator. The regulator links these IDs with the company's personnel list to validate voters' eligibility. Subsequently, a layer of cryptographic security is added as the Private Share Keys (PSK) for each voter is generated and distributed. These keys are encrypted to enhance security. Voters also generate their public and private key pairs, providing them with cryptographic tools for secure voting. The voter list of public key addresses is then published and ensures that only verified individuals can participate in the election.
- **Voting:** The Voting stage prioritises voter privacy and election integrity. Voters choose their preferred candidates and record their selections in a transaction. These transactions are signed using a traceable ring signature, thus preserving anonymity and allowing for the real-time tracking of candidate preferences. These signed ballots and their hash values are broadcast to the blockchain network for verification. The integrity of the process is upheld as signatures are validated and votes are authenticated.
- **Tallying:** The election results are meticulously calculated and disclosed in the Tallying and Announcing stage. Verifiers access the blockchain to count the ballots, leveraging the number and content of transactions to determine the outcome. Results are compared against predefined requirements for accuracy. The consensus mechanism, typically PBFT, guarantees the consistency of results. Once consensus is achieved, the official election results are announced and permanently recorded on the blockchain.

Security Analysis:

- **Strengths:** The protocol ultimately ensures voter anonymity, enables auditability through traceable ring signatures, and maintains fairness and correctness in elections. Furthermore, The integration of certificates and code-based cryptography introduces notable advantages. This implementation enables the detection of voter misconduct, thereby enhancing the system's robustness. Additionally, using code-based cryptography provides resistance against potential quantum computing attacks, addressing a significant concern in long-term cryptographic security.
- **Weaknesses:** The protocol has limitations in terms of scalability, making it best suited for elections with a limited number of participants [149]. It may need to fully meet the accuracy, scalability, and voter variability requirements. Additionally, depending on the scale of the election, some efficiency may be sacrificed when trying to enhance privacy protections [150].

7) Self-Tallying Voting System in Decentralised IoT

The authors in [128] argued for a comprehensive framework for a Blockchain-based self-voting system that applies decentralisation in decentralised Internet of Things (IoT) software

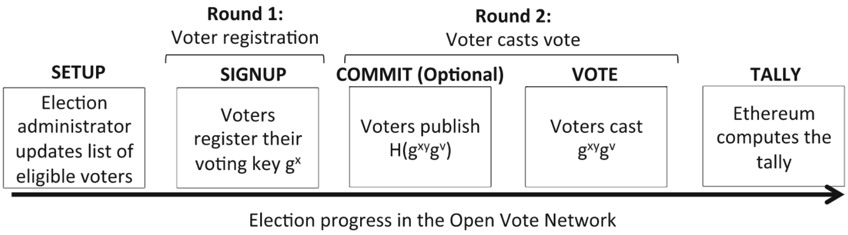
Round 1:
Voter registration
Round 2:
Voter casts vote
SETUP
Election administrator updates list of eligible voters
SIGNUP
Voters register their voting key g^x
COMMIT (Optional)
Voters publish $H(g^{xy}g^v)$
VOTE
Voters cast $g^{xy}g^v$
TALLY
Ethereum computes the tally
Election progress in the Open Vote Network

FIGURE 5. The five stages to the Open Vote Network Election [126].

updates.

Technical Specifications:

- **Setup:** All voting machines must register with the IoT gateway upon enrolment to obtain an intranet IP address. Each voting machine generates a public/private key pair using this IP address as a random seed and publishes the public key to the Blockchain. The machines then compute a simple ZKP with the private key to prove public key ownership and post it on the Blockchain.
- **Voting:** To maintain privacy, each voting machine randomises its selected vote with its private key and other machines' public keys to generate a secret ballot. After that, each machine submits its vote to the Blockchain.
- **Tallying:** In the tally phase, after the ballots from all the voting machines have been recorded on the blockchain, any authorised entity can retrieve and gather these ballots. Subsequently, the final voting result can be computed by removing the hiding factor associated with each secret ballot.

Security Analysis:

- **Strengths:** The proposed framework offers a multitude of advantages. The blockchain overlay enables complete decentralisation, thereby mitigating the potential vulnerability of a single point of failure. The protocol ensures fairness by employing the principles of commitment, homomorphic encryption, and ZKP. In addition, the system can calculate results automatically when users follow the instructions. This feature guarantees a secure voting process by utilising encrypted ballots that satisfy the criterion of indistinguishability. The system's structure also allows software updates via the blockchain. This enhances its ability to adapt and survive long-term.
- **Weaknesses:** Although the system has many benefits, it has limitations. The primary weakness lies in its potential for high execution costs, mainly when the implementation requires support from many IoT devices. This scalability issue could prevent widespread adoption, especially when budget constraints are a significant consideration. Additionally, IoT devices are susceptible to various security threats, which raises significant security concerns [151]. Cost is another factor, as IoT devices can be expensive to purchase, install, and maintain. Furthermore, interoperability challenges may arise, as IoT devices are not always compatible with one.

8) AGORA

[129] introduced the AGORA commercial setup comprising four layers: a bulletin board, Catena, the Bitcoin Blockchain, and Votapp. This setup provides a voting mechanism that allows for auditing election results at any point in the voting process and enables anyone to watch an election. **Technical Specifications:**

- **Setup:** Before the start of an election, administrators create a configuration file containing parameters unique to that event. The parameters include essential elements like the identities of the responsible officials, the eligible voters, the anonymisation nodes, the designated time frame for the casting phase (commencement and conclusion), the specific type of election being conducted, and the comprehensive list of candidates available for selection.
- **Voting:** During the voting phase, the voter's ballot is encrypted using the ElGamal cryptosystem and distributed consensus nodes of the Agora network known as the Cothority. The encrypted ballot is then posted on the Bulletin Board, and a Cothority node receives and authenticates it.
- **Tallying:** After the voting phase, Agora's network processes all ballots through a mixing network using the Neff shuffling method and the ZKP to generate a new list of anonymous ballots. The anonymised ballots are collectively decrypted and published on the bulletin board, and the Cothority nodes provide proof of decryption correctness to execute the tallying phase. During the tallying phase, administrators use partially decrypted votes that have been handled correctly to reconstruct the original plaintext ballots that were previously anonymised. The plaintext ballots are displayed on the Notice Board to facilitate tallying and votes are counted across all legitimate decrypted ballots after the decryption phase. The results are then posted on the Bulletin Board by Agora nodes. Each stage of the election is described in Figure 6:

Security Analysis:

- **Strengths:** Agora offers a technologically advanced and user-friendly platform that enhances elections' security, transparency, and efficiency.
- **Weaknesses:** Dependence on outside observers creates a possible security threat, as collusion with candidates may compromise the integrity of the results [152]. These shortcomings emphasise the need for further research and development to strengthen the overall security and reliability of the Agora system.

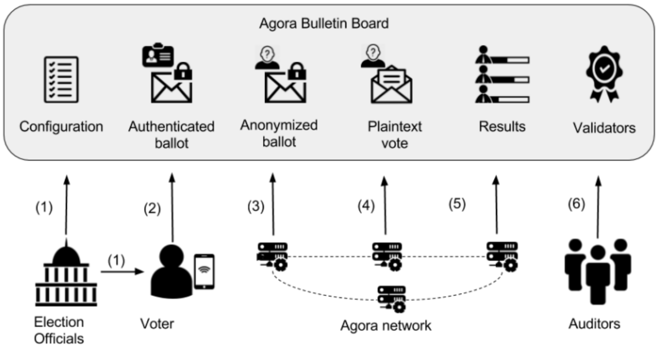

FIGURE 6. *Agora Voting Process [129].*

9) e-voting System Using an Enterprise Blockchain

[131] outlines an enterprise blockchain-based approach to secure e-voting and addresses common system challenges.

Technical Specifications:

- **Setup:** The e-voting system setup begins with a security parameter (κ) and generates group parameters (q, G, G, g, g, e) and the election authority's key pair (`SK_A`, `PK_A`) using `KeyGen(`κ`)`. During the voting stage, voters register by presenting official credentials, generating their public-private key pair (PK_i, SK_i), and submitting their public key. The authenticated public keys and user IDs form the list of eligible voters. In the `Register(`i, $\mathcal{R}$`)` process, pre-registered users send a commitment C_i derived from a secret value (s_i) masked with a random value (r_i), both unknown to the voter. The voter signs C_i using SK_i to create a signature (σ_i), and the registrar ($\mathcal{R}$) posts an entry $\langle i, C_i, \sigma_i \rangle$ in a public database, which can be verified by matching the public key to the certified voter list.
- **Voting:** Registered voters can anonymously cast their ballots during the voting phase (TVoting). Each voter reveals their secret number s used in their commitment C_i and proves knowledge of the secret value$r$. This is achieved through a zkSNARK construction, utilising a pairing system for encoding computations as Quadratic Arithmetic programmes (QAPs). The zkSNARK proof is a cryptographic mechanism for proving knowledge of s and r without revealing their values. The Randomiser generates fresh encryption keys (K) for each vote and computes values such as g^s and h_j (where j ranges from 0 to $\kappa - 1$), based on stored values of s and r. These computed values and vote choices (v) are used as input to the proof algorithm to generate the zkSNARK proof π. The zkSNARK proof acts as a signature for the encrypted vote $EK(v)$, incorporating random values the prover uses. Once the proof is created, values r and h_j are permanently erased for security, while K and s are retained for the next phase. The ballot (b) is formed, including $EK(v)$, s, and π, and is posted anonymously to the blockchain.
- **Tallying and Verification Stage:** Following the post-voting phase (TPostVoting), where encryption keys are posted to the blockchain, the tallying and verification process begins. Keys are matched to encrypted votes, enabling the counting process. The tallying process is straightforward, with all necessary information available on the blockchain. Talliers or any interested parties can conduct the tallying process. To ensure transparency and verify the tally (τ), the `VerifyTally` function can be used. Decryption is facilitated by storing encrypted votes in a hashtable, with the value s as the key. This efficient storage approach allows for $O(1)$ decryption per vote or $O(n)$ overall.

Security Analysis:

- **Strengths:** The proposed voting system offers robustness against coercion, effectively protecting the integrity of the voting process even when voters are pressured to reveal their choices. Furthermore, the system's efficiency in tallying votes is notable, as there is a linear overhead about the number of voters, thus ensuring a practical and streamlined counting process.
- **Weaknesses:** The system's absence of real-world testing raises concerns about its practical usefulness and robustness in the face of unexpected challenges. Further analysis is needed to determine the financial implications, legal compliance, and usability of the election authorities' reliance on trust. This is crucial to ensure the system's viability and widespread adoption among varied voter groups.
  Time Consuming and Costly

10) ProvotuMN

A permission Distributed Ledger (DL) system -ProvotuMN 3.0 is a decentralised and receipt-free (RF) voting system based on an end-to-end verifiable Re-Encryption Mixnet (RMN). Using cryptographic shuffles, NIZKP, and distributed key generation, ProvotuMN decentralises trust and ensures RF. Performance evaluations demonstrate its scalability for large-scale voting [132].

Technical Specifications:

- **Setup:** The sealers generate DL credentials during the pre-voting phase. Following this, they register themselves with the Verifying Authority (VA). Subsequently, the VA takes charge of bootstrapping the DL and establishing the Public Bulletin Board (PBB). The execution

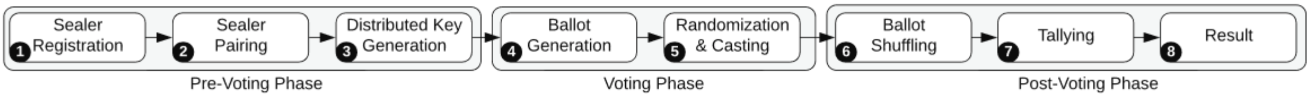

FIGURE 7. *ProvotuMN's Voting Scheme.*

of the Distributed Key Generation (DKG) protocol follows a decentralised approach and guarantees that the process is distributed across multiple participants.

Voting: During the voting phase, voters exercise their right to choose by selecting their preferred options. They then proceed to create their respective ballots. To enhance the level of privacy, an additional layer of security is applied to the ballots through re-encryption and utilises a Randomiser (R). Subsequently, the voters submit their re-encrypted ballots to the decentralised Public Bulletin Board (DPBB). This DPBB serves as the designated platform responsible for collecting and decentralising the submitted ballots' storage.

Tallying: After the voting phase, the VA ends the voting process, which subsequently triggers the start of the post-voting phase. During this phase, the submitted ballots undergo a decentralised randomisation process in which their order is shuffled to ensure the secrecy and integrity of the votes. A collaborative decryption procedure is executed following the random mixing to convert the encrypted ballots into their respective plaintext representations. This collaborative decryption uses the combined efforts of multiple participants to ensure a fair and transparent outcome. Finally, the decrypted ballots are tallied to determine the final voting results and provide an accurate representation of the collective preferences indicated by the voters

Security Analysis:

- **Strengths:** The ProvotuMN 3.0 voting systems offer notable advantages over traditional systems. Implementing the flexible vote encoding feature improves transparency and reduces the computing workload on voters, leading to a more user-friendly system. Furthermore, ProvotuMN guarantees anonymity by distributing trust across mixers, utilising cryptographic methods like shuffling and re-encryption.
- **Weaknesses:** The system's complexity and runtime result in an increase in correlation with the number of mix servers and sealers, potentially impacting overall performance. Although ProvotuMN achieves Ballot Secrecy and Receipt Freeness, it lacks Coercion Resistance. This limitation may compromise ballot secrecy in collusion between Randomisers and voters, exposing voters to potential coercion, manipulation, or the disclosure of their credentials. Furthermore, the proposed system cannot be deployed on Ethereum as it requires its permissiond blockchain [153].

11) AvecVoting

The authors in [133] propose a protocol that uses blockchain technology to tackle key issues in modern e-voting.

Technical Specifications:

- **Setup:** The initiator sets up a single-choice vote using smart contracts, specifying parameters such as the topic, number of options, deadlines, total voters and counters, minimum secret shares for threshold key restoration, and revenue and fines for counters. Voters register with the initiator by providing their identity and ring public key, and the initiator maintains a list of registered voters ($Lvoter$). Off-chain, voters engage in key negotiation, generating and securely sharing secret shares crucial for decryption during the counting stage.
- **Voting:** During the voting stage, voters choose their preferred option from the list of voting options and encrypt their ballot using a threshold public key generated from all shared public parameters. To ensure anonymity, voters create one-time ring signatures on their ballots. The encrypted ballots and their corresponding signatures are then submitted to the Vote Main smart contract for secure and confidential vote recording.
- **Tallying and Verification:** In the tallying and verification stage, voters submit their secret shares to the Vote Main smart contract, after which no further submissions are accepted. The counter committee (CC) is formed through Random Sortition, and each counter in CC calculates a threshold private key using the submitted secret shares to decrypt the ballots. Counters verify each ballot using one-time ring signatures and decrypt them using the threshold private key. Valid votes are counted for each voting option, and counting vectors are generated for each counter. The final tallying is performed by collecting and aggregating counting results from all counters. Honest counters are rewarded, and those with incorrect results are penalised through the PayOff contract, reinforcing the system's integrity.

Security Analysis:

- **Strengths:** AvecVoting employs blockchain technology to enhance the security of e-voting systems, improving the protection of anonymity, privacy, and accuracy. The system emphasises preserving voters' privacy and enhancing the ability to verify ballots, increasing transparency and building trust independently. AvecVoting's PayOff mechanism effectively incentivises accuracy and shows security under challenging conditions.
- **Weaknesses:** The proposed system uses smart contracts developed in Solidity, which run on the EVM. However,

this approach faces challenges, as complex operations on the EVM can be very expensive. Additionally, Layer 1 solutions still have a low transaction rate of around 15 transactions per second. As a result, the system may become inefficient as elections grow in scale [154].

12) Scalable Self-Tallying Blockchain-Based Voting

SBvote is influenced by the work of BBB-Voting [155]. The system maintains a central authority for registration and organising while including decentralisation to improve scalability [134].

Security Analysis:

- **Strengths:** During this phase, participants create ephemeral key pairs comprising private and public keys, which are crucial for the privacy of their votes. These ephemeral keys are used to compute Multi-Party Computation (MPC) keys without revealing individual key values. Furthermore, to enhance privacy, participants are assigned to voting groups in a pseudo-random manner. This pseudo-random assignment reduces the likelihood of unanimous votes within groups, potentially compromising ballot secrecy. The setup phase establishes a robust cryptographic framework and a randomised grouping strategy to protect voters' privacy and set the stage for secure e-voting.
- **Voting Phase:** To maintain the confidentiality of voting choices, voters cast their ballots as blinded votes. These blinded votes are created by combining their ephemeral private keys with the generator corresponding to their chosen candidate. Importantly, each blinded vote is accompanied by a NIZK proof of set membership, which ensures that the vote correctly encrypts a valid candidate's generator without revealing the voter's choice. This phase safeguards voters' privacy while allowing them to participate actively in the electoral process.
- **Tallying Phase:** In the tallying phase, the SBvote protocol ensures the accuracy and integrity of the election results. Leveraging the self-tallying property, any interested party can verify the correctness of tallies without revealing individual votes. The tally computation involves an exhaustive search for a solution to an equation with numerous combinations. This challenging mathematical task requires efficient methods for tally aggregation and ensures that the results accurately reflect the voters' choices. The self-tallying feature and verification mechanisms guarantee transparency and fairness in the election outcome, and the tallying phase marks the culmination of the SBvote protocol, delivering trustworthy election results to the participating voters and external observers alike.

Security Analysis:

- **Strengths:** SBvote boasts several notable advantages, including its scalability through decentralised organisation, robust privacy provisions, security measures, and public verifiability. The system maintains perfect ballot secrecy and offers both individual and universal verifiability, instilling confidence in the integrity of the voting process.
- **Weaknesses:** SBvote faces challenges in handling unanimous votes due to the potential privacy risks associated with associating IP addresses, which could compromise the anonymity of voters. Additionally, choosing the correct platform parameters for large-scale elections is challenging due to the system's high computational demands and transaction costs. Its dependence on the throughput of the underlying blockchain also restricts its ability to manage large-scale voting, despite attempts to enhance scalability [156].

VII. COMPARISON OF SECURITY AND PRIVACY CHARACTERISTICS

Table 2 shows the comparative examination of the e-voting systems environment. The evaluation encompasses key characteristics such as privacy, verifiability, double voting prevention, coercion mitigation and scalability. The present study serves as an academic foundation for assessing e-voting schemes, as this supports future research and facilitates policy discussions in this domain.

A. PRIVACY

Privacy in e-voting systems ensures voters can cast their ballots without fear of revealing their choices. A review of various schemes reveals diverse approaches to achieving privacy. For instance, [116] system employs strong cryptographic measures such as the DDH assumption and secure ZKP primitives. On a different note, [117] strategy focuses on security against polynomial time adversaries and utilises simulated NIZK. However, a potential drawback in their Centralised setting is the risk of exposing voter choices. On the other hand, [134] adopt strategies like limiting adversary control and transaction batching to prioritise the protection of individual vote privacy within voting groups. Similarly, [118] supports implementing a two-step encryption process with AES and RSA to safeguard intercepted voting data from being associated with specific voters. Further, [122] protocol uses blind signatures and hash functions to create a challenge in linking voters to their ballots, particularly when votes are cast randomly. This protocol acknowledges the risk of IP address exposure through blockchain network communication and suggests using anonymity services like TOR to boost privacy. In blockchain-based systems, [123] support incorporating blockchain technology, secret sharing, and Paillier's homomorphic encryption to protect voter anonymity and data transmission privacy. An additional layer of security is achieved through oblivious transfer and allows receivers access to only selected messages. Meanwhile, [128] system ensures conditional privacy with ballots accessible only to a coalition of a subset of voting machines. Also, [129] ensures voter privacy through verifiable ballot encryption and anonymisation by employing threshold ElGamal encryption and Neff shuffling. Moreover, [130] system also focuses on deterring

Target Feature	References	Proposed Strategy
Privacy	[116]	Uses DDH assumption and ZKP for vote secrecy.
	[118]	Employs AES and RSA encryption for data protection.
	[134]	Focuses on adversary control limits and transaction batching.
	[117]	Implements NIZK proofs, with Centralised risk.
	[122]	utilises blind signatures, hash functions and suggests TOR for anonymity.
	[123]	Combines blockchain, secret sharing, and Paillier encryption.
	[128]	Offers conditional privacy with coalition-accessible ballots.
	[129]	Agora system uses threshold ElGamal encryption and Neff shuffling.
	[130]	Uses anonymous channels and zkSNARKs to prevent vote linkage.
	[124]	Implements blind signatures and Idemix for voter anonymity.
	[133]	AvecVoting system with threshold encryption and ring signatures.
	[131]	Voter privacy is ensured via cryptographic identities in a SoftHSM.
Verifiability	[116]	Uses ZKP for key and value verification on public boards.
	[117]	Enhances transparency with digital signatures and NIZK.
	[121]	zero watermarking and encryption.
	[134]	Implements self-tallying for verifiability.
	[118]	Provides comprehensive vote storage and counting verifiability.
	[122]	Ensures auditable voting via blockchain.
	[123]	utilises blockchain for ballot and result verification.
	[128]	Offers post-voting self-tallying for verifiability.
	[129]	Uses public blockchain for full voting process verifiability.
	[130]	Prevents double voting with cryptographic commitments on blockchain.
	[124]	Employs blockchain ledger for transparent vote verification.
	[131]	Verifiability is achieved with unique transaction IDs for each vote.
Prevent Double Voting	[116]	Uses tokens and digital signatures for immutable records.
	[117]	Records voters once a ballot is cast.
	[134]	Likely has similar double voting safeguards.
	[118]	Employs a real-time digital voter list and NFC tags.
	[122]	Uses blockchain to detect and count unique votes.
	[124]	Allows voting only for registered public keys.
	[129]	utilises Consensus Nodes for transaction confirmation.
	[130]	Implies blockchain and token-based authentication prevent double voting.
	[131]	Double voting is prevented using NFTs as ballots linked to voter identities.
Mitigates Coercion	[116]	Uses private booths and signed transcripts to deter coercion.
	[117]	Focuses on physical security at polling stations.
	[134]	Implies voter autonomy consideration (specific strategies not detailed).
	[118]	Employs supervised booths, staff oversight, and omits voting receipts.
	[119]	utilises single voting with return codes, enhancing voter autonomy.
	[121]	Allows multiple voting; only the last ballot is valid.
	[124]	Suggests using ring signatures; lacks receipt-freeness.
	[129]	Does not detail specific anti-coercion mechanisms.
	[130]	Employs a Randomiser token for ballot construction to prevent coercion.
	[132]	Achieves receipt-freeness, preventing vote verification to third parties.
	[133]	Focuses on voter anonymity and complex voting process to deter coercion.
	[131]	Coercion is indirectly mitigated by anonymous transactions and cryptographic identities.
Scalability	[116]	Designs for scalability with parallelisable cryptographic processes.
	[117]	Demonstrates strong scalability using efficient computation and storage for ballots and NIZK proofs.
	[134]	Chooses platforms like Gnosis and Harmony for low costs and high throughput.
	[118]	Suggests scalability with efficient data management and real-time updates.
	[121]	Shows potential scalability with fast cryptographic operations.
	[126]	Faces scalability challenges on Ethereum, indicating the need for alternatives.
	[129]	Encounters scalability challenges due to its blockchain network structure.
	[130]	Prepared for large-scale elections with scalable blockchain technology.
	[132]	Suitable for large-scale voting with decentralised approach and efficient cryptography.
	[124]	utilises Hyperledger Fabric for its scalability over public blockchains.
	[133]	Designed for scalability with efficient off-chain counting and algorithms.
	[131]	Scalability is achieved through Hyperledger Fabric's efficient transaction architecture

TABLE 2. Comparison of e-voting systems based on key features.

TABLE 3. Assessing Voting Schemes: A Comprehensive Evaluation of Key Factors

#	Schemes	Approach	Privacy	Universal verifiability	Prevent Double Voting	Mitigates Coercion	Scalability
1	[116]	DRE	✓	✓	✓	✓	✓
2	[117]	DRE	✓	✓	✓	✓	✓
3	[122]	Blockchain	✓	✓	✓	Not discussed	Not discussed
4	[123]	Blockchain	✓	✓	Not discussed	Not discussed	Not discussed
5	[126]	Blockchain	Not discussed	✓	✓	✗	✗
6	[129]	Blockchain	✓	✓	✓	Not discussed	✗
7	[121]	Internet voting	Not discussed	✓	Not discussed	✓	✓
8	[124]	Blockchain	✓	✓	✓	✗	✓
9	[134]	Blockchain	✓	✓	Not discussed	Not discussed	✓
10	[128]	Blockchain	✓	✓	Not discussed	Not discussed	Not discussed
11	[130]	Blockchain	✓	✓	✓	✓	✓
12	[118]	DRE	✓	✓	✓	✓	✓
13	[131]	Blockchain	✓	✓	✓	✓	✓
14	[132]	Blockchain	✓	Not discussed	Not Discussed	✓	✓
15	[133]	Blockchain	✓	Not discussed	Not discussed	✓	✓

the linkage of votes to individual voters. It achieves this using anonymous channels for vote submission and advanced cryptographic methods like zkSNARKs. Similarly, [124] methodology employs cryptographic solutions like blind signatures and identity mixers (Idemix) to anonymise voter identities and choices, thereby precluding any direct correlation of votes to specific individuals. [133] AvecVoting system is noteworthy for its use of threshold encryption and one-time ring signatures that offers significant anonymity and ensuring that the voter's identity and vote remain untraceable. Finally, [131] system places emphasis on voter privacy by utilising cryptographic identities stored within a software emulation of a Hardware Security Module (SoftHSM). This mechanism plays a critical role in maintaining the confidentiality and security of voting actions.

B. UNIVERSAL VERIFIABILITY:

Universal verifiability allows voters and stakeholders to validate election results objectively. This principle is seen in various systems and diverse approaches. For example, [116] system enables any observer to verify the integrity of keys and computational values using ZKP on a public bulletin board. Also, [117] methodology further bolsters transparency with digital signatures and NIZK proofs on their bulletin board to make the election process more transparent and verifiable. On another front, [134] highlights self-tallying properties that enable individual and universal verifiability. This feature allows any party to verify booth tallies and adds a layer of assurance to the voting process. The system in [118] offers individual verifiability where voters can be assured that their votes have been correctly stored and counted. Furthermore, it ensures global verifiability by accounting for all votes and ensuring only eligible votes are counted. Also, [121] ensures the verifiability of votes and robust voter identification by employing advanced techniques such as zero watermarking and encryption. In blockchain-based solutions, [122] protocol makes the voting process auditable and transparent to all participants by recording it on the blockchain. Similarly, [123] system leverages blockchain technology for data verifiability and allows transparent and independent verification of ballots and election results by voters. Furthermore, [128] system introduces a unique mechanism that enables self-tallying after all voting machines have cast their ballots, which allows any entity in the system to perform the tallying. This feature significantly contributes to universal verifiability, ensuring the fairness and accuracy of the election results. The Agora system takes transparency to another level with its public blockchain, the Bulletin Board, where all voting data is stored and made available for verification [129]. This openness allows for full public verifiability of the entire voting process. Moreover, [130] effectively combats double voting by utilising unique cryptographic commitments and Randomisers for each voter. This ensures that everyone can only vote once as these commitments are securely recorded and validated on the blockchain. The approach presented in [131] further strengthens verifiability by assigning a unique transaction ID to each vote to track and confirm that votes have not been changed. This mechanism enhances the reliability of the voting process as each vote is both traceable and cannot be modified without being detected. Finally, the blockchain's transparent and immutable ledger significantly enhances the verifiability in [124]. This ledger records all votes and allows anyone to verify the results and confirm that each vote is accurately counted.

C. PREVENTION OF DOUBLE VOTING

Preventing double voting is essential for maintaining the integrity of e-voting systems. Various studies, including those by [116], have employed unique tokens, serial numbers, and digital signatures to establish an immutable voting record. Moreover, [117] indicates that a voter is recorded as having voted once a ballot is cast, thereby preventing subsequent votes. While [134] needs to provide detailed information on their approach, similar protection mechanisms are likely already in place. In addressing this challenge, [118] utilises a digital voter list that is updated in real-time to prevent individuals from voting more than once. Furthermore, this digital list, in conjunction with unlocking tokens such as NFC tags, ensures the uniqueness of each vote. Moreover, [122] emphasises the prevention of double voting through blockchain network security measures, ensuring that if two ballots have identical voting strings, they are counted only once. An alternative approach suggested by [124] allows a voter to cast a vote only if their identifier (public key) is among the registered keys for the election, thereby ensuring unreusability. In contrast, [129] prevents double voting through its set of Consensus Nodes, which confirm transactions on the Bulletin Board and maintain the integrity of the voting process. Additionally, [130] implies that the prevention of double voting is inherent in its system due to the properties of blockchain technology and the unique token-based system for voter authentication, suggesting the inclusion of mechanisms that ensure each voter can only vote once. Conversely, [131] ensures the prevention of double voting using non-fungible tokens (NFTs) as ballots minted by an authorised organisation and tied to specific voter identities. This innovative approach leverages the distinct properties of NFTs to guarantee that each vote is unique and associated with a single voter, effectively preventing any possibility of double voting.

D. MITIGATES COERCION

Mitigating coercion is essential for ensuring freedom and fairness in electoral processes. However, many existing systems do not adequately address coercion mitigation. According to the schemes proposed by [116], private voting booths and digitally signed commitment transcripts are used to deter coercion. On the other hand, [117] focuses on the physical security of polling stations to prevent coercion. In contrast, the strategies for coercion mitigation in the system presented by [134] are not explicitly mentioned. Nevertheless, the overall design of their system suggests consideration for voter autonomy. In the case of [118], coercion is mitigated through traditional election controls such as supervised voting booths and staff oversight and by not providing voting receipts, thus reducing the potential for coercion and vote-selling. The system discussed in the paper by [119] may indirectly contribute to reducing coercion through single voting with return codes, allowing voters to verify their votes without revealing their choices. This makes it more difficult for external parties to exert influence or verify the casting vote, enhancing voter autonomy and reducing the risk of coercion. Additionally, the SeVEP system, as described by [121], enhances coercion resistance by allowing multiple voting within a specific period, where only the last ballot cast is considered valid. While [124] currently lacks receipt-freeness, it suggests using ring signatures to mitigate coercion. However, [129] does not explicitly detail specific mechanisms to mitigate coercion. The approach to coercion resistance by [130] is vital to their system, accomplished by utilising a Randomiser token – a tamper-resistant source of randomness – for ballot construction, ensuring that voters cannot be coerced into voting a certain way or demonstrating how they voted. [132] achieves receipt-freeness in their system, meaning voters cannot obtain information that could be used to prove to a third party how they voted, significantly diminishing the risk of coercion. On the other hand, the AvecVoting system, outlined in the paper by [133], indirectly assists in mitigating coercion through its design that guarantees voter anonymity and prevents linking votes to individual voters. Although the system does not specify explicit anti-coercion strategies, its focus on voter privacy and the complexity of the voting process serves as deterrents against coercion, contributing to the overall resilience against coercive tactics in voting. Finally, [131] does not explicitly discuss how the system mitigates coercion. However, privacy and security measures, such as anonymous transactions and cryptographic identities, could indirectly help reduce coercion by protecting voter identities and choices.

E. SCALABILITY

The evaluation of e-voting schemes critically relies on their scalability, which determines their ability to handle an increasing number of voters effectively. Several schemes, such as those proposed by [116], are specifically designed for scalability, incorporating parallelisable cryptographic processes suitable for large electorates. For instance, the system developed by [117] demonstrates robust scalability by utilising efficient computational and storage mechanisms for ballot generation and non-interactive zero-knowledge (NIZK) proofs. Furthermore, the choice of platforms, such as Gnosis and Harmony, by [134] aligns with objectives for low operational costs and high throughput, reflecting their commitment to scalability. On the other hand, while [118] does not explicitly address scalability, its efficient data management and real-time updates suggest a design capable of accommodating more significant voter numbers and more complex voting scenarios. In the SeVEP system proposed by [121], efficiency is apparent in handling complex cryptographic operations within an average time of less than 6.50 seconds per voter during both pre-polling and polling phases, which indicates potential scalability. In contrast, [126], which relies on the Ethereum blockchain, faces scalability issues due to transaction throughput and block size constraints, emphasising the need for enhancements or alternative blockchain solutions for large-scale elections. Similarly, [129] encounters scalability challenges due to its blockchain network structure consisting of write-permission and read-only nodes, highlighting the necessity for advancements to support large-scale elections efficiently. Addressing these challenges, [130] harnesses blockchain technology with efficient cryptographic techniques, ensuring that vote tallying is linearly scalable and practical for widespread implementation. Additionally, the system developed by [132] employs a decentralised approach and efficient cryptographic operations to manage a significant number of votes without performance degradation, showcasing its potential for scalability in high-volume electoral scenarios. Moreover, the system developed by [124] utilises Hyperledger Fabric, a permissioned blockchain platform known for its scalability advantages over public blockchains like Ethereum or Bitcoin. This strategic choice facilitates the efficient handling of a substantial volume of votes. Using Hyperledger Fabric's architecture for efficient transaction processing and confirmation, the platform's execute-order-validate architecture enhances scalability compared to traditional blockchain systems. Additionally, [131] addresses scalability using Hyperledger Fabric's architecture, allowing for efficient transaction processing and confirmation. The platform's execute-order-validate architecture enhances scalability compared to traditional blockchain systems. Lastly, [133] is designed explicitly with scalability in mind. Performance evaluations indicate its capability to meet the demands of real-world elections with acceptable overheads. Implementing counters for off-chain counting, alongside the RandomSortition and reputation-based PayOff algorithms, significantly contributes to its ability to process a high volume of votes efficiently.

VIII. DISCUSSION, LIMITATIONS, AND FUTURE WORK

When evaluating the three main methodologies for e-voting, including Internet voting, DRE systems, and blockchain-based systems, it becomes apparent that the blockchain approach emerges as the most viable option. Blockchain based schemes excel in privacy protection, universal verifiability, prevention of double voting, and coercion mitigation, bolstering the integrity and transparency of the electoral process. While they may present higher complexity, their robust security features make them a compelling option for modern e-voting systems. In contrast, while providing some transparency, Internet voting raises concerns about privacy and coercion mitigation, and DRE systems, although offering moderate privacy and verification capabilities, still grapple with scalability issues. Thus, blockchain-based systems emerge as the most balanced and effective choice among the three approaches for ensuring secure, transparent, and trustworthy electronic elections. However, this research is subject to some restrictions and concerns that provide opportunities for future exploration. The scalability of the e-voting system is a significant challenge, as a blockchain-based system with a smaller number of voters is more cost-effective than one with a larger number of voters, which results in increased transaction confirmation times. Scalability remains a crucial factor to investigate in the cost analysis of blockchain-based e-voting systems.

IX. CONCLUSION

This survey provides a comprehensive overview of e-voting systems, containing cryptographic techniques, blockchain technologies, security models, and evaluation frameworks. It highlights the growing significance of blockchain in e-voting and offers a structured framework for understanding the underlying terminology and elements. The study categorises and explains key concepts required for blockchain-based e-voting systems, including consensus algorithms, frameworks, performance evaluation metrics, and cryptographic tools. By presenting an updated overview of the current blockchain e-voting system, whether implemented by governments and companies or proposed by academics, this survey facilitates a clear understanding of the diverse e-voting landscape. This survey highlights the need to continue advancing research and innovation by identifying e-voting systems' challenges, including preserving privacy, scalability, and reducing coercion. It highlights how the reviewed systems address these challenges and points towards research directions that can contribute to developing trustworthy and widely accepted e-voting systems.

REFERENCES

[1] M. I. Shamos, "Paper v. electronic voting records-an assessment," in *Proceedings of the 14th ACM Conference on Computers, Freedom and Privacy*, pp. 1–23, 2004.

[2] U. O. Ekong and V. Ekong, "M-voting: a panacea for enhanced e-participation," *Asian Journal of Information Technology*, vol. 9, no. 2, pp. 111–116, 2010.

[3] R. Anane, R. Freeland, and G. Theodoropoulos, "e-voting requirements and implementation," in *The 9th IEEE International Conference on E-Commerce Technology and The 4th IEEE International Conference on Enterprise Computing, E-Commerce and E-Services (CEC-EEE 2007)*, pp. 382–392, 2007.

[4] O. Cetinkaya and D. Cetinkaya, "Verification and validation issues in electronic voting," *Electronic journal of e-government*, vol. 5, no. 2, pp. pp117–126, 2007.

[5] G. Ofori-Dwumfuo and E. Paatey, "The design of an electronic voting system," *Research Journal of Information Technology*, vol. 3, no. 2, pp. 91–98, 2011.

[6] S. Nakamoto and A. Bitcoin, "A peer-to-peer electronic cash system," *Bitcoin.–URL: https://bitcoin. org/bitcoin. pdf*, vol. 4, no. 2, p. 15, 2008.

[7] B. S. White, C. G. King, and J. Holladay, "Blockchain security risk assessment and the auditor," *Journal of Corporate Accounting & Finance*, vol. 31, no. 2, pp. 47–53, 2020.

[8] L. Chi and X. Zhu, "Hashing techniques: a survey and taxonomy," *ACM Computing Surveys*, vol. 50, no. 1, pp. 1–36, 2017.

[9] A. Darwish and M. M. El-Gendy, "A new cryptographic voting verifiable scheme for e-voting system based on bit commitment and blind signature," *Int J Swarm Intel Evol Comput*, vol. 6, no. 158, p. 2, 2017.

[10] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[11] G. Jain, "Digital signature algorithm," *International Journal of Innovations in Computing*, vol. 1, no. 1, pp. 1–6, 2012.

[12] W. J. Caelli, E. P. Dawson, and S. A. Rea, "Pki, elliptic curve cryptography, and digital signatures," *Computers & Security*, vol. 18, no. 1, pp. 47–66, 1999.

[13] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE transactions on information theory*, vol. 31, no. 4, pp. 469–472, 1985.

[14] V. G. Martinez, L. Hernández-Álvarez, and L. H. Encinas, "Analysis of the cryptographic tools for blockchain and bitcoin," *Mathematics*, vol. 8, no. 1, p. 131, 2020.

[15] K. T. N. Swe, *Providing Data Integrity and Authentication using RSA Digital Signature and MD5 Hash Algorithm.* PhD thesis, MERAL Portal.

[16] S. Xinglin and S. Fei, "A two-party collaborative blind signature scheme based on sm9," in *2024 7th International Conference on Artificial Intelligence and Big Data (ICAIBD)*, pp. 288–295, IEEE, 2024.

[17] A. Fujioka, T. Okamoto, and K. Ohta, "A practical secret voting scheme for large scale elections," in *International Workshop on the Theory and Application of Cryptographic Techniques*, pp. 244–251, Springer, 1992.

[18] Y. Kho, S. H. Heng, and J. J. Chin, "A review of cryptographic electronic voting," *Symmetry*, vol. 14, no. 5, p. 858, 2022.

[19] S. Ibrahim, M. Kamat, M. Salleh, and S. Aziz, "Secure e-voting with blind signature," in *4th National Conference of Telecommunication Technology, 2003. NCTT 2003 Proceedings.*, pp. 193–197, 2003.

[20] M. Chang, I. Chen, I. Wu, and Y. Yeh, "Schnorr blind signature based on elliptic curves," *Asian Journal of Information Technology, Published by Grace Publication Network*, pp. 130–134, 2003.

[21] R. Haenni and O. Spycher, "Secure internet voting on limited devices with anonymized {DSA} public keys," in *2011 Electronic Voting Technology Workshop/Workshop on Trustworthy Elections (EVT/WOTE 11)*, 2011.

[22] A. A. Thu and K. T. Mya, "Implementation of an efficient blind signature scheme," *International journal of innovation, management and technology*, vol. 5, no. 6, p. 443, 2014.

[23] I. Jabbar and S. Najim, "Using fully homomorphic encryption to secure cloud computing," *Internet of Things and Cloud Computing*, vol. 4, no. 2, pp. 13–18, 2016.

[24] P. V. Parmar, S. B. Padhar, S. N. Patel, N. I. Bhatt, and R. H. Jhaveri, "Survey of various homomorphic encryption algorithms and schemes," *International Journal of Computer Applications*, vol. 91, no. 8, 2014.

[25] I. Miers, C. Garman, M. Green, and A. D. Rubin, "Zerocoin: anonymous distributed e-cash from bitcoin," in *2013 IEEE Symposium on Security and Privacy*, pp. 397–411, IEEE, 2013.

[26] S. Lee, H. Ko, J. Kim, and H. Oh, "vcnn: Verifiable convolutional neural network based on zk-snarks," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–17, 2023.

[27] E. Ben-Sasson, A. Chiesa, E. Tromer, and M. Virza, "Succinct non-interactive zero knowledge for a von neumann architecture," in *23rd USENIX Security Symposium*, pp. 781–796, USENIX Association, 2014.

[28] R. Lavin, X. Liu, H. Mohanty, L. Norman, G. Zaarour, and B. Krishnamachari, "A survey on the applications of zero-knowledge proofs," *arXiv preprint arXiv:2408.00243*, 2024.

[29] A. Diro, L. Zhou, A. Saini, S. Kaisar, and P. C. Hiep, "Leveraging zero knowledge proofs for blockchain-based identity sharing: A survey of advancements, challenges and opportunities," *Journal of Information Security and Applications*, vol. 80, p. 103678, 2024.

[30] E. Ben-Sasson, I. Bentov, Y. Horesh, and M. Riabzev, "Scalable, transparent, and post-quantum secure computational integrity," *Cryptology ePrint Archive*, 2018.

[31] S. Deng and B. Du, "zkTree: A zero-knowledge recursion tree with ZKP membership proofs." Cryptology ePrint Archive, Paper 2023/208, 2023.

[32] R. Bögli, *Assessing RISC Zero using ZKit: An Extensible Testing and Benchmarking Suite for ZKP Frameworks*. PhD thesis, OST Ostschweizer Fachhochschule, 2024.

[33] M. Ambrona, A.-L. Schmitt, R. R. Toledo, and D. Willems, "New optimization techniques for plonk's arithmetization." Cryptology ePrint Archive, Paper 2022/462, 2022. https://eprint.iacr.org/2022/462.

[34] L. Grassi, D. Khovratovich, C. Rechberger, A. Roy, and M. Schofnegger, "Poseidon: A new hash function for {Zero-Knowledge} proof systems," in *30th USENIX Security Symposium (USENIX Security 21)*, pp. 519–535, 2021.

[35] M. Albrecht, L. Grassi, C. Rechberger, A. Roy, and T. Tiessen, "Mimc: Efficient encryption and cryptographic hashing with minimal multiplicative complexity," in *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 191–219, Springer, 2016.

[36] L. Grassi, D. Khovratovich, R. Lüftenegger, C. Rechberger, M. Schofnegger, and R. Walch, "Reinforced concrete: A fast hash function for verifiable computation," in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1323–1335, 2022.

[37] K. Gurkan, K. W. Jie, and B. Whitehat, "Community proposal: Semaphore: Zero-knowledge signaling on ethereum," *Accessed: Jul 2020*, vol. 1, p. 2021.

[38] K.-H. Wang, S. K. Mondal, K. Chan, and X. Xie, "A review of contemporary e-voting: requirements, technology, systems and usability," *Data Science and Pattern Recognition*, vol. 1, no. 1, pp. 31–47, 2017.

[39] R. Taş and O. O. Tanrı"over, "A systematic review of challenges and opportunities of blockchain for e-voting," *Symmetry*, vol. 12, no. 8, p. 1328, 2020.

[40] X. Xu, C. Pautasso, L. Zhu, V. Gramoli, A. Ponomarev, A. B. Tran, and S. Chen, "The blockchain as a software connector," in *2016 13th Working IEEE/IFIP Conference on Software Architecture (WICSA)*, pp. 182–191, 2016.

[41] E. J. Scheid, B. B. Rodrigues, C. Killer, M. F. Franco, S. Rafati, and B. Stiller, "Blockchains and distributed ledgers uncovered: clarifications, achievements, and open issues," in *Advancing Research in Information and Communication Technology: IFIP's Exciting First 60+ Years, Views from the Technical Committees and Working Groups*, pp. 289–317, Springer, 2021.

[42] S. Zhang, K. Zheng, and B. Wang, "A v2v electricity transaction scheme with privacy protection based on the internet of vehicles and consortium blockchain," *International Journal of Electrical Power & Energy Systems*, vol. 157, p. 109789, 2024.

[43] T. T. A. Dinh, J. Wang, G. Chen, R. Liu, B. C. Ooi, and K.-L. Tan, "Blockbench: A framework for analyzing private blockchains," in *Proceedings of the 2017 ACM international conference on management of data*, pp. 1085–1100, 2017.

[44] I. Karamitsos, M. Papadaki, and N. B. Al Barghuthi, "Design of the blockchain smart contract: A use case for real estate," *Journal of Information Security*, vol. 9, no. 3, pp. 177–190, 2018.

[45] M. Touloupou, K. Christodoulou, A. Inglezakis, E. Iosif, and M. Themistocleous, "Towards a framework for understanding the performance of blockchains," in *2021 3rd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)*, pp. 47–48, 2021.

[46] T. T. A. Dinh, R. Liu, M. Zhang, G. Chen, B. C. Ooi, and J. Wang, "Untangling blockchain: A data processing view of blockchain systems," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 7, pp. 1366–1385, 2018.

[47] S. Mollajafari and K. Bechkoum, "Blockchain technology and related security risks: Towards a seven-layer perspective and taxonomy," *Sustainability*, vol. 15, no. 18, p. 13401, 2023.

[48] Q. T. Thai, N. Ko, S. H. Byun, and S. Kim, "Design and implementation of ndn-based ethereum blockchain," *Journal of Network and Computer Applications*, vol. 200, p. 103329, 2022.

[49] A. C. Naik, A. M. Prajapati, S. N. Pandey, and A. C. Mishra, "Blockchain based e-voting system," in *2023 7th International Conference on Trends in Electronics and Informatics (ICOEI)*, pp. 316–320, 2023.

[50] S. Wang, Y. Yuan, X. Wang, J. Li, R. Qin, and F.-Y. Wang, "An overview of smart contract: Architecture, applications, and future trends," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, pp. 108–113, 2018.

[51] X. Ye and M. König, "From the graphical representation to the smart contract language: a use case in the construction industry," in *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, vol. 38, pp. 272–279, IAARC Publications, 2021.

[52] W. Metcalfe *et al.*, "Ethereum, smart contracts, dapps," *Blockchain and Crypt Currency*, vol. 77, 2020.

[53] A. Benahmed Daho, "Crypto-spatial: an open standards smart contracts library for building geospatially enabled decentralized applications on the ethereum blockchain," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 43, pp. 421–426, 2020.

[54] S. Suratkar, M. Shirole, and S. Bhirud, "Cryptocurrency wallet: A review," in *2020 4th International Conference on Computer, Communication and Signal Processing (ICCCSP)*, pp. 1–7, 2020.

[55] C. Fan, S. Ghaemi, H. Khazaei, and P. Musilek, "Performance evaluation of blockchain systems: A systematic survey," *IEEE Access*, vol. 8, pp. 126927–126950, 2020.

[56] A. K. Yadav and M. Garg, "Docker containers versus virtual machine-based virtualization," in *Emerging Technologies in Data Mining and Information Security: Proceedings of IEMIS 2018, Volume 3*, pp. 141–150, Springer, 2019.

[57] S. El Kafhali, "Blockchain-based electronic voting system: Significance and requirements," *Mathematical Problems in Engineering*, vol. 2024, no. 1, p. 5591147, 2024.

[58] "Ethereum virtual machine (evm) opcodes." *EtherVM*. [Online]. Available: https://www.ethervm.io/, Accessed: Sep. 4, 2024.

[59] A. Y. Wong, E. G. Chekole, M. Ochoa, and J. Zhou, "On the security of containers: Threat modeling, attack analysis, and mitigation strategies," *Computers & Security*, vol. 128, p. 103140, 2023.

[60] J. Rosa-Bilbao, J. Boubeta-Puig, J. Lagares-Galán, and M. Vella, "Leveraging complex event processing for monitoring and automatically detecting anomalies in ethereum-based blockchain networks," *Computer Standards & Interfaces*, p. 103882, 2024.

[61] S. Seibold and G. Samman, "Consensus: Immutable agreement for the internet of value," *KPMG< https://assets. kpmg. com/content/dam/kpmg/pdf/2016/06/kpmgblockchain-consensus-mechanism. pdf*, 2016.

[62] F. Liu, H.-Y. Fan, and J.-Y. Qi, "Blockchain technology, cryptocurrency: entropy-based perspective," *Entropy*, vol. 24, no. 4, p. 557, 2022.

[63] M. Pawlak and A. Poniszewska-Marańda, "Trends in blockchain-based electronic voting systems," *Information Processing & Management*, vol. 58, no. 4, p. 102595, 2021.

[64] B. Singhal, G. Dhameja, and P. S. Panda, *Beginning Blockchain: A Beginner's guide to building Blockchain solutions*, vol. 1. Springer, 2018.

[65] X. Xu, I. Weber, M. Staples, L. Zhu, J. Bosch, L. Bass, C. Pautasso, and P. Rimba, "A taxonomy of blockchain-based systems for architecture design," in *2017 IEEE international conference on software architecture (ICSA)*, pp. 243–252, IEEE, 2017.

[66] M. Pawlak and A. Poniszewska-Marańda, "Trends in blockchain-based electronic voting systems," *Information Processing & Management*, vol. 58, no. 4, p. 102595, 2021.

[67] H. F. Ouattara, D. Ahmat, F. T. Ouédraogo, T. F. Bissyandé, and O. Sié, "Blockchain consensus protocols," in *International Conference on e-Infrastructure and e-Services for Developing Countries*, pp. 304–314, Springer, 2017.

[68] A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder, *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*. Princeton University Press, 2016.

[69] A. Kiayias, A. Russell, B. David, and R. Oliynykov, "Ouroboros: A provably secure proof-of-stake blockchain protocol," in *Annual international cryptology conference*, pp. 357–388, Springer, 2017.

[70] P. He, D. Tang, and J. Wang, "Staking pool centralization in proof-of-stake blockchain network," *Available at SSRN 3609817*, 2020.

[71] M. Castro, B. Liskov, *et al.*, "Practical byzantine fault tolerance," in *OsDI*, vol. 99, pp. 173–186, 1999.

[72] P. Zhang and M. Zhou, "Security and trust in blockchains: Architecture, key technologies, and open issues," *IEEE Transactions on Computational Social Systems*, vol. 7, no. 3, pp. 790–801, 2020.

[73] S. Zhai, Y. Yang, J. Li, C. Qiu, and J. Zhao, "Research on the application of cryptography on the blockchain," in *Journal of Physics: Conference Series*, vol. 1168, p. 032077, IOP Publishing, 2019.

[74] F. Z. Chentouf and S. Bouchkaren, "Security and privacy in smart city: a secure e-voting system based on blockchain," *International Journal of Electrical and Computer Engineering*, vol. 13, no. 2, p. 1848, 2023.

[75] V. Buterin *et al.*, "A next-generation smart contract and decentralized application platform," *White Paper*, vol. 3, no. 37, pp. 2–1, 2014.

[76] H. Sáez de Ocáriz Borde, "An overview of trees in blockchain technology: Merkle trees and merkle patricia tries," 02 2022.

[77] C. Yue, Z. Xie, M. Zhang, G. Chen, B. C. Ooi, S. Wang, and X. Xiao, "Analysis of indexing structures for immutable data," in *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pp. 925–935, 2020.

[78] J. Kuszmaul, "Verkle trees," tech. rep., MIT, 2018. MIT, Tech. Rep. Available: https://math.mit.edu/research/highschool/primes/materials/2018/Kuszmaul.pdf.

[79] C. Sguanci, R. Spatafora, and A. M. Vergani, "Layer 2 blockchain scaling: A survey," *arXiv preprint arXiv:2107.10881*, 2021.

[80] A. Gangwal, H. R. Gangavalli, and A. Thirupathi, "A survey of layer-two blockchain protocols," *Journal of Network and Computer Applications*, vol. 209, p. 103539, 2023.

[81] L. T. Thibault, T. Sarry, and A. S. Hafid, "Blockchain scaling using rollups: A comprehensive survey," *IEEE Access*, vol. 10, pp. 93039–93054, 2022.

[82] C. F. Torres, A. Mamuti, B. Weintraub, C. Nita-Rotaru, and S. Shinde, "Rolling in the shadows: Analyzing the extraction of mev across layer-2 rollups," *arXiv preprint arXiv:2405.00138*, 2024.

[83] A. Gangwal, H. R. Gangavalli, and A. Thirupathi, "A survey of layer-two blockchain protocols," *Journal of Network and Computer Applications*, vol. 209, p. 103539, 2023.

[84] X. Tang, L. Shi, A. Lai, Y. Du, J. Deng, J. Fu, and J. Li, "Smart contract migration: Security analysis and recommendations from ethereum to arbitrum," *arXiv preprint arXiv:2307.14773*, 2023.

[85] I. Roșca, A.-I. Butnaru, and E. Simion, "Security of ethereum layer 2s," 2023.

[86] Optimism Foundation, "Optimism: an Ethereum-compatible layer-2 blockchain." Available: https://www.optimism.io/, 2024. Version 1.1.4.

[87] S. Li, M. Liu, and M. Chen, "Omnilytics+: A secure, efficient, and affordable blockchain data market for machine learning through off-chain processing," *arXiv preprint arXiv:2406.06477*, 2024.

[88] F. Chard and C. Fletcher-Smith, "Blockchain scalability for smart contract systems using eutxo model," *arXiv preprint arXiv:2202.00561*, 2022.

[89] T. Lavaur, J. Detchart, J. Lacan, and C. P. Chanel, "Modular zk-rollup on-demand," *Journal of Network and Computer Applications*, vol. 217, p. 103678, 2023.

[90] F. Chard and C. Fletcher-Smith, "Blockchain scalability for smart contract systems using eutxo model," *arXiv preprint arXiv:2202.00561*, 2022.

[91] F. Bruschi, D. Sciuto, T. Paulon, and A. Marchesi, "A decentralized approach to award game achievements," in *2023 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, pp. 237–242, IEEE, 2023.

[92] O. O. Okediran and R. A. Ganiyu, "A framework for electronic voting in nigeria," *International Journal of Computer Applications*, vol. 129, no. 3, pp. 12–16, 2015.

[93] S. Weaver, *Machines of democracy: How voting technology threatens the integrity of elections in the United States*. Texas Woman's University, 2005.

[94] M. Germann and U. Serdült, "Internet voting for expatriates: The swiss case," *eJournal of eDemocracy & Open Government*, vol. 6, no. 2, pp. 197–215, 2014.

[95] M. F. Mursi, G. M. Assassa, A. Abdelhafez, and K. M. A. Samra, "On the development of electronic voting: a survey," *International Journal of Computer Applications*, vol. 61, no. 16, 2013.

[96] P. Baudier, G. Kondrateva, C. Ammi, and E. Seulliet, "Peace engineering: The contribution of blockchain systems to the e-voting process," *Technological Forecasting and Social Change*, vol. 162, p. 120397, 2021.

[97] J. Huang, D. He, M. S. Obaidat, P. Vijayakumar, M. Luo, and K.-K. R. Choo, "The application of the blockchain technology in voting systems: A review," *ACM Computing Surveys (CSUR)*, vol. 54, no. 3, pp. 1–28, 2021.

[98] M. Volkamer, J. Mylopoulos, N. M. Sadeh, M. J. Shaw, C. Szyperski, and W. Aalst, *Evaluation of electronic voting: requirements and evaluation procedures to support responsible election authorities*, vol. 30. Springer, 2009.

[99] R. Anane, R. Freeland, and G. Theodoropoulos, "E-voting requirements and implementation," in *The 9th IEEE International Conference on E-Commerce Technology and The 4th IEEE International Conference on Enterprise Computing, E-Commerce and E-Services (CEC-EEE 2007)*, pp. 382–392, IEEE, 2007.

[100] M. Volkamer, J. Mylopoulos, N. M. Sadeh, M. J. Shaw, C. Szyperski, and W. Aalst, *Evaluation of electronic voting: requirements and evaluation procedures to support responsible election authorities*, vol. 30. Springer, 2009.

[101] H. Alamleh and A. A. S. AlQahtani, "Analysis of the design requirements for remote internet-based e-voting systems," in *2021 IEEE World AI IoT Congress (AIIoT)*, pp. 0386–0390, IEEE, 2021.

[102] K.-H. Wang, S. K. Mondal, K. Chan, and X. Xie, "A review of contemporary e-voting: Requirements, technology, systems and usability," *Data Science and Pattern Recognition*, vol. 1, no. 1, pp. 31–47, 2017.

[103] O. M. Olaniyi, O. T. Arulogun, and E. O. Omidiora, "Design of secure electronic voting system using multifactor authentication and cryptographic hash functions," *IOSR Journal of Computer Engineering (IOSR-JCE)*, vol. 17, pp. 86–97, November–December 2015. e-ISSN: 2278-0661, p-ISSN: 2278-8727.

[104] L. Fouard, M. Duclos, and P. Lafourcade, "Survey on electronic voting schemes," *supported by the ANR project AVOTÉ*, 2007.

[105] A. Schneider, C. Meter, and P. Hagemeister, "Survey on remote electronic voting," *arXiv preprint arXiv:1702.02798*, 2017.

[106] S. Mello-Stark and E. A. Lamagna, "The need for audit-capable e-voting systems," in *2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, pp. 535–540, IEEE, 2017.

[107] B. M. B. Pereira, J. M. Torres, P. M. Sobral, R. S. Moreira, C. P. d. A. Soares, and I. Pereira, "Blockchain-based electronic voting: A secure and transparent solution," *Cryptography*, vol. 7, no. 2, p. 27, 2023.

[108] A. Schneider, C. Meter, and P. Hagemeister, "Survey on remote electronic voting," *arXiv preprint arXiv:1702.02798*, 2017.

[109] F. Shirazi, S. Neumann, I. Ciolacu, and M. Volkamer, "Robust electronic voting: Introducing robustness in civitas," in *2011 International Workshop on Requirements Engineering for Electronic Voting Systems*, pp. 47–55, IEEE, 2011.

[110] M. Hajian Berenjestanaki, H. R. Barzegar, N. El Ioini, and C. Pahl, "Blockchain-based e-voting systems: a technology review," *Electronics*, vol. 13, no. 1, p. 17, 2023.

[111] S. El Kafhali, "Blockchain-based electronic voting system: Significance and requirements," *Mathematical Problems in Engineering*, vol. 2024, no. 1, p. 5591147, 2024.

[112] S. J. Laskowski, M. Autry, J. Cugini, W. Killam, and J. Yen, *Improving the usability and accessibility of voting systems and products*. US Department of Commerce, National Institute of Standards and Technology, 2004.

[113] C.-K. Wu and R. Sankaranarayana, "Internet voting: concerns and solutions," in *First International Symposium on Cyber Worlds, 2002. Proceedings.*, pp. 261–266, IEEE, 2002.

[114] D. Petcu and D. A. Stoichescu, "A hybrid mobile biometric-based e-voting system," in *2015 9th International Symposium on Advanced Topics in Electrical Engineering (ATEE)*, pp. 37–42, IEEE, 2015.

[115] M. Volkamer, O. Spycher, and E. Dubuis, "Measures to establish trust in internet voting," in *Proceedings of the 5th International Conference on Theory and Practice of Electronic Governance*, ICEGOV '11, (New York, NY, USA), p. 1–10, Association for Computing Machinery, 2011.

[116] F. Hao, M. N. Kreeger, B. Randell, D. Clarke, S. F. Shahandashti, and P. H.-J. Lee, "Every vote counts: ensuring integrity in large-scale electronic voting," in *2014 Electronic Voting Technology Workshop/Workshop on Trustworthy Elections*, pp. 1–10, USENIX Association, 2014.

[117] S. Bag, M. A. Azad, and F. Hao, "E2e verifiable borda count voting system without tallying authorities," in *Proceedings of the 14th International Conference on Availability, Reliability and Security*, pp. 1–9, 2019.

[118] V. Agate, A. De Paola, P. Ferraro, G. Lo Re, and M. Morana, "Secureballot: a secure open source e-voting system," *Journal of Network and Computer Applications*, vol. 191, p. 103165, 2021.

[119] D. Galindo, S. Guasch, and J. Puiggali, "2015 neuchâtel's cast-as-intended verification mechanism," in *International Conference on E-Voting and Identity*, pp. 3–18, Springer, 2015.

[120] S. Kardas, M. S. Kiraz, M. A. Bing"ol, and F. Birinci, "Norwegian internet voting protocol revisited: ballot box and receipt generator are allowed to collude," *Security and Communication Networks*, vol. 9, no. 18, pp. 5051–5063, 2016.

[121] A. Qureshi, D. Meg'ıas, and H. Rif'a-Pous, "Sevep: secure and verifiable electronic polling system," *IEEE Access*, vol. 7, pp. 19266–19290, 2019.

[122] Y. Liu and Q. Wang, "An e-voting protocol based on blockchain," Report 2017/1043, Cryptology ePrint Archive, 2017.

[123] J.-H. Hsiao, R. Tso, C.-M. Chen, and M.-E. Wu, "Decentralized e-voting systems based on the blockchain technology," in *Advances in Computer Science and Ubiquitous Computing*, pp. 305–309, Springer, 2017.

[124] D. Kirillov, V. Korkhov, V. Petrunin, M. Makarov, I. M. Khamitov, and V. Dostov, "Implementation of an e-voting scheme using hyperledger fabric permissioned blockchain," in *International Conference on Computational Science and Its Applications*, pp. 509–521, Springer, 2019.

[125] S. Bartolucci, P. Bernat, and D. Joseph, "Sharvot: secret share-based voting on the blockchain," in *Proceedings of the 1st International Workshop on Emerging Trends in Software Engineering for Blockchain*, pp. 30–34, 2018.

[126] P. McCorry, S. F. Shahandashti, and F. Hao, "A smart contract for boardroom voting with maximum voter privacy," in *International Conference on Financial Cryptography and Data Security*, pp. 357–375, Springer, 2017.

[127] S. Gao, D. Zheng, R. Guo, C. Jing, and C. Hu, "An anti-quantum e-voting protocol in blockchain with audit function," *IEEE Access*, vol. 7, pp. 115304–115316, 2019.

[128] G. Han, Y. Li, Y. Yu, K.-K. R. Choo, and N. Guizani, "Blockchain-based self-tallying voting system with software updates in decentralized iot," *IEEE Network*, vol. 34, no. 4, pp. 166–172, 2020.

[129] N. Gailly, P. Jovanovic, B. Ford, J. Lukasiewicz, and L. Gammar, "Agora: bringing our voting systems into the 21st century," white paper, Agora Voting, 2018.

[130] T. Dimitriou, "Efficient, coercion-free and universally verifiable blockchain-based voting," *Computer Networks*, vol. 174, p. 107234, 2020.

[131] C. Denis González, D. Frias Mena, A. Massó Muñoz, O. Rojas, and G. Sosa-Gómez, "Electronic voting system using an enterprise blockchain," *Applied Sciences*, vol. 12, no. 2, p. 531, 2022.

[132] C. Killer, M. Eck, B. Rodrigues, J. von der Assen, R. Staubli, and B. Stiller, "Provotumn: decentralized, mix-net-based, and receipt-free voting system," in *2022 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pp. 1–9, IEEE, 2022.

[133] M. Li, X. Luo, W. Sun, J. Li, and K. Xue, "Avecvoting: Anonymous and verifiable e-voting with untrustworthy counters on blockchain," in *ICC 2022 - IEEE International Conference on Communications*, pp. 4751–4756, 2022.

[134] I. Stančíková and I. Homoliak, "Sbvote: Scalable self-tallying blockchain-based voting," in *Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing*, pp. 203–211, 2023.

[135] S. Panja and B. Roy, "A secure end-to-end verifiable e-voting system using blockchain and cloud server," *Journal of Information Security and Applications*, vol. 59, p. 102815, 2021.

[136] S. Panja, S. Bag, F. Hao, and B. Roy, "A smart contract system for decentralized borda count voting," *IEEE Transactions on Engineering Management*, vol. 67, no. 4, pp. 1323–1339, 2020.

[137] V. Agate, P. Ferraro, G. L. Re, and S. K. Das, "Blind: A privacy preserving truth discovery system for mobile crowdsensing," *Journal of Network and Computer Applications*, vol. 223, p. 103811, 2024.

[138] J. Müller and T. Truderung, "A protocol for cast-as-intended verifiability with a second device," *arXiv preprint arXiv:2304.09456*, 2023.

[139] K. Gjøsteen, "The norwegian internet voting protocol," in *International Conference on E-Voting and Identity*, pp. 1–18, Springer, 2011.

[140] M. Cavinkumaran, A. Srinivasan, and K. Vijayakumar, "Hyperledger iroha for a secure and efficient voting system," in *2024 2nd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT)*, pp. 380–386, IEEE, 2024.

[141] G. Rathee, R. Iqbal, O. Waqar, and A. K. Bashir, "On the design and implementation of a blockchain enabled e-voting application within iot-oriented smart cities," *IEEE Access*, vol. 9, pp. 34165–34176, 2021.

[142] R. Clarke, L. McGuire, M. Baza, A. Rasheed, and M. Alsabaan, "Online voting scheme using ibm cloud-based hyperledger fabric with privacy-preservation," *Applied Sciences*, vol. 13, no. 13, p. 7905, 2023.

[143] A. Chavan, A. Jadhav, S. Chandre, S. Rathod, R. Bhende, and D. Patil, "Revolutionizing voting: Blockchain-powered e-voting with solana," in *2024 1st International Conference on Trends in Engineering Systems and Technologies (ICTEST)*, pp. 1–6, IEEE, 2024.

[144] M. N. Razzaque, K. M. R. Alam, M. A. Habib, and M. T. Hasan, "Enhancing e-voting security with blockchain-backed decentralized authorization," in *2023 6th International Conference on Electrical Information and Communication Technology (EICT)*, pp. 1–6, IEEE, 2023.

[145] U. Jafar, M. J. Ab Aziz, Z. Shukur, and H. A. Hussain, "A systematic literature review and meta-analysis on scalable blockchain-based electronic voting systems," *Sensors*, vol. 22, no. 19, p. 7585, 2022.

[146] M. Pawlak and A. Poniszewska-Marańda, "Trends in blockchain-based electronic voting systems," *Information Processing & Management*, vol. 58, no. 4, p. 102595, 2021.

[147] U. Jafar, M. J. A. Aziz, and Z. Shukur, "Blockchain for electronic voting system—review and open research challenges," *Sensors*, vol. 21, no. 17, p. 5874, 2021.

[148] Z. Li, S. Majumdar, and E. Pournaras, "Blockchain-based decentralized time lock machines: Automated reveal of time-sensitive information," *arXiv preprint arXiv:2401.05947*, 2024.

[149] K. Jain, M. Singh, H. Gupta, and A. Bhat, "Quantum resistant blockchain-based architecture for secure medical data sharing," in *2024 3rd International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*, pp. 1400–1407, IEEE, 2024.

[150] A. Singh, A. Ganesh, R. R. Patil, S. Kumar, R. Rani, and S. K. Pippal, "Secure voting website using ethereum and smart contracts," *Applied System Innovation*, vol. 6, no. 4, p. 70, 2023.

[151] P. Mwansa and B. Kabaso, "Blockchain electoral vote counting solutions: A comparative analysis of methods, constraints, and approaches," in *2023 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD)*, pp. 1–10, IEEE, 2023.

[152] P. Shelke, S. Dedgaonkar, N. Gopale, N. Deogaonkart, and N. Joshi, "Block chain-based e-voting system using smart contract," *AITC-2023 and CSSP-2023*, p. 55, 2023.

[153] R. Muth and F. Tschorsch, "Tornado vote: Anonymous blockchain-based voting," in *2023 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pp. 1–9, IEEE, 2023.

[154] A. de Castro and C. Coutinho, "Electronic voting through blockchain: A survey," in *2023 5th International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, pp. 1–6, IEEE, 2023.

[155] S. Venugopalan, I. Homoliak, Z. Li, and P. Szalachowski, "Bbb-voting: 1-out-of-k blockchain-based boardroom voting," *arXiv preprint arXiv:2010.09112*, 2020.

[156] U. Jafar, M. J. Ab Aziz, Z. Shukur, and H. A. Hussain, "Empowering secure and cost-efficient blockchain electronic voting by optimized zk-snark algorithm," in *2023 International Conference on Electrical Engineering and Informatics (ICEEI)*, pp. 1–6, IEEE, 2023.

MOSBAH ALOWN received a B.Sc. degree in computer science from Al Albayet University in 2015 and an M.Sc. degree in the same field from the same university in 2019. He is pursuing a PhD at De Montfort University, Leicester, U.K., focusing on e-voting. Alongside his doctoral studies, he holds a significant role as a part-time lecturer at the cybersecurity institution at De Montfort University. His research interests lie primarily in cybersecurity and blockchain technology, particularly developing and analysing e-voting systems.

MEHMET SABIR KIRAZ received his BSc from the Mathematics Department at Middle East Technical University, Ankara, Turkey, in 2000. He then worked as a software engineer with the IT Departments of Pamukbank and Yapi Kredi Bank between June 2000 and September 2002. In 2003, he obtained his MSc (with a full scholarship) from the International Max Planck Research Institute for Computer Science, Germany. His thesis title is "*Formalisation and Verification of Informal Security Protocol Description*". Following this, he pursued his PhD from the Computer Science Department at Eindhoven Technical University, Eindhoven, The Netherlands, which he completed in 2008. His thesis title is "*Secure and Fair Two-Party Computation*". Between 2008 and 2010, he worked as a test coordinator with PHILIPS in Eindhoven and TOMTOM in Amsterdam. From January 2010 to November 2018, he worked as a chief researcher at TUBITAK BILGEM Informatics and Information Security Research centre. In 2017, he established and served as the director of the Blockchain Research Lab. In the same year, he was also promoted to the rank of associate professor by the Turkish Inter-University Council. Currently, he holds the position of senior lecturer at De Montfort University, UK, where he has been working since November 2018. His current research interests are cryptography, cryptographic protocols, privacy, PKI, authentication and key management, secure multiparty computation, homomorphic encryption, e-voting, cloud security, VANETs, blockchain, and cryptocurrencies.

MUHAMMED ALI BINGOL received a BSc degree in Telecommunications Engineering and an MS degree in Electronics and Communication Engineering from Istanbul Technical University, Istanbul, Turkey, in 2008 and 2012, respectively. He earned his PhD degree in Computer Science and Engineering from Sabancı University, Istanbul, Turkey, in 2019. During his master's studies, he was a visiting scientist at the Université Catholique de Louvain, Belgium. He worked as a research engineer at the Telecommunications Software and Systems Group (TSSG), Ireland, in 2007. From 2008 to 2020, he served as a Chief Researcher at TÜBİTAK BİLGEM National Research Institute of Electronics and Cryptology. Currently, he holds the position of Senior Lecturer in Cyber Security at De Montfort University, United Kingdom. His primary research interests include cryptographic protocols, secure multi-party computation, private function evaluation, blockchain security & privacy, authentication systems, and e-voting.

• • •