electronics

MDPI

Article

Distributed Anonymous e-Voting Method Based on Smart Contract Authentication

Wenjie Tang [1], Wenzhong Yang [1,2,*], Xiaodan Tian [1] and Shaoqi Yuan [1]

[1] School of Information Science and Engineering, Xinjiang University, Urumqi 830046, China; twj@stu.xju.edu.cn (W.T.); lianlian@stu.xju.edu.cn (X.T.); 107552103609@stu.xju.edu.cn (S.Y.)
[2] Xinjiang Key Laboratory of Multilingual Information Technology, Xinjiang University, Urumqi 830046, China
* Correspondence: yangwenzhong@xju.edu.cn

Abstract: With the development of science and technology, the traditional centralized ballot management will lead to the risk of an opaque voting process and tampering of back-end data, and it can no longer meet the requirements of e-voting transparency and anonymity, while the distributed blockchain technology, with its features of being transparent and tamper-proof, can well solve the problems in traditional e-voting and make the ballot firmly in the hands of users. To make each node reach consensus, each node can access the complete shared ledger, and malicious nodes can analyze the transaction information in the ledger to obtain users' voting information, which makes anonymity impossible to guarantee. We propose a blockchain technology-based voting record synchronization model and an anonymous authentication model, using zk-SNARK and Merkle tree technology, that achieve user authentication and anonymous voting. In this scheme, the user's random identity address is inserted into the Merkle tree, and when voting, it is only necessary to prove that the user's identity address is in the tree to complete anonymous voting without revealing the user's real identity. This scheme meets the basic requirements of electronic voting and ensures the reliability and security of voting.

Keywords: Ethereum; privacy protection; voting; zero-knowledge proof

check for updates

Citation: Tang, W.; Yang, W.; Tian, X.; Yuan, S. Distributed Anonymous e-Voting Method Based on Smart Contract Authentication. *Electronics* **2023**, *12*, 1968. https://doi.org/10.3390/electronics12091968

Academic Editors: Mehdi Sookhak

Received: 19 March 2023
Revised: 17 April 2023
Accepted: 18 April 2023
Published: 24 April 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The right to vote is the foundation of a democratic society and represents the "empowerment" of the individual, through which citizens can participate in elections and express their views on professional life, cities, and government through elections, referenda, or surveys [1]. Voting in elections has gone from "hand" to "ballot", from "ballot" to "machine", and from "machine" to electronic. Voting by ballot, instead of by hand, ensures the expression of voters' true opinions to a certain extent, but it is less efficient, and has fewer security, privacy, fraud, integrity, and privacy issues. With the development of computers, voting is replaced by "machine voting", which does not require human participation in vote counting and improves the efficiency of voting. However, there are some risks in "machine voting", the background data can be easily tampered with, and the transparency and anonymity of the voting process cannot be guaranteed. With the rapid development of internet technology, many traditional offline services have been transferred to online, and online voting is called e-voting, where the users of e-voting are voters and election authorities. The voter can submit their ballot to the election authority through e-voting anytime, anywhere, and the election authority is responsible for collecting the voter's ballot. Due to the advantages of electronic voting, such as time-saving, efficiency, and flexibility, it is gaining more and more attention and gradually replacing traditional voting methods [2]. However, unlike traditional voting methods, electronic voting systems may suffer from system failures, and network and information security issues.

E-voting is becoming increasingly popular because it has a great potential to efficiently handle large numbers of voters and save time and costs. Existing e-voting systems aim not

only to improve the accessibility of voting but also to maintain a secure and tamper-proof system [3], and many countries such as Australia, Brazil, Estonia, the Netherlands, and Norway have adopted e-voting in national elections. However, these attempts to implement e-voting protocols and systems show that some security issues remain unresolved [4,5]. For example, during the 2020 presidential election in the United States, the electronic voting system Dominion Voting Systems was accused of the risk of ballot data tampering. In the 2021 Albanian parliamentary elections, ballot data was tampered with and led to disputed election results. There were reports that hackers attacked an election counting center in the country, to change the election results by tampering with ballot information, among other things.

In 2008, Satoshi Nakamoto first introduced the concept and technology of blockchain-Bitcoin in his paper [6], laying the technical foundation for distributed voting, which is a public ledger that can operate without a central authority. To ensure data integrity, all nodes on the blockchain verify and store each transaction. Users create transactions, which are then collected by "miners" into blocks. For the miner to attach their block to the blockchain, they must complete proof, such as proof of employment or proof of equity. Due to the blockchain's append-only structure and the computational power required to add blocks to the chain, at least 51% of the network's computational power is colluded to rewrite part of the blockchain record. Due to these properties, blockchains are considered immutable and secure data structures.

In December 2013, Vitalik Buterin proposed the Ether blockchain platform, which applies smart contract technology to the blockchain, extending this functionality by implementing smart contracts. Enabling blockchain to be used in areas other than digital currency, makes the Ether platform a good candidate for distributed voting. First, the Ethereum platform does not require a trusted third party, and an electronic voting system can be easily constructed using a server (typically a trusted third party); however, servers are subject to single-point-of-failure problems, that result in data loss, but blockchain replaces centralized servers with a set of independent nodes, thereby eliminating the single-point-of-failure problems associated with centralized voting. Second, the hashing and linking of blocks in the blockchain make it mathematically impossible to modify or tamper with voting data. Third, Ethereum allows smart contracts to be written and enforced by all participating nodes, guaranteeing their accuracy [7]. Smart contracts are blocks of code that are stored on the blockchain. Smart contracts consist of functions or events that allow contracts to interact with each other and users. Since the smart contract code is deployed on the blockchain, the code cannot be modified and is only available to the nodes connected to the blockchain. To protect the system from malicious users, and to compensate miners for the use of computing power, the execution of each transaction includes a transaction fee, called "GAS" in Ether. GAS is a unit of measurement for the amount of work done by an operation, and the price of GAS is calculated in Ether.

With the in-depth study of blockchain, the drawbacks of blockchain are becoming apparent. Although blockchain contains many security measures, due to its distributed consensus nature, attackers can still analyze network traffic and transaction information in the public ledger and thus gain access to the information of voting users [8]. Cryptographic data storage methods have the potential to solve these problems. Usually, cryptography is used to encrypt the data before it is stored on the server, to encrypt voter data, voting results, etc., so this method can ensure the authenticity and security of voting information [9]. Currently, researchers have integrated various schemes into distributed voting systems and have improved e-voting to varying degrees, thereby increasing the anonymity of people participating in the voting as well as reducing the cost of development.

In this work, we use a combination of non-interactive zero-knowledge proof and Merkle tree technology for anonymous authentication, and to ease the computational pressure on the blockchain, we place the proof generation off-chain and then only verify on-chain. To ensure the efficiency of the whole system, we use the non-interactive zero-

knowledge proof algorithm Groth16. Finally, the web3js technology is used for on-chain and off-chain communication.

2. Related Work

The main purpose of this paper is to use blockchain technology to provide secure electronic voting to ensure true democracy, which lies in ensuring that all citizens have an equal right to choose the true candidate they have in mind in an unbiased manner. Ballot tampering, voter coercion, etc., defy these ideas and defeat the purpose of voting in the first place [10–12]. Since the introduction of blockchain technology, many researchers have focused on blockchain technology as the underlying technology for e-voting, with some using permission chains as the underlying technology for their research on e-voting, while others have chosen unpermitted chains as their technology. Smart contracts extend the use of permissioned blockchains; in contrast to non-permissioned blockchains, permissioned blockchains can only be accessed by one organization. This enhances the privacy of the blockchain while sacrificing some of its decentralized properties. Therefore, most of the research on applications on Ethernet has focused on permission chains, Table 1 summarizes the three types of blockchains. The rest of this section summarizes the literature related to blockchain-based e-voting and the methods used in recent years.

Table 1. Blockchain classification.

Types	Descriptions	Consensus Rate	Scenes
Public Chain	Anyone can participate and can be accessed globally	Slow	Global dispersion scenarios
TAlliance Chain	Controlled by pre-selected nodes within the federation	Fast	Enterprises in selected organizations
Private Chain	Write access is controlled by the organization	Fast	Information sharing and management in the organization

In [13], a method was proposed for voting using blockchain and Paillier encryption. It exploits the tamper-proof property of blockchain and the additive homomorphism property of Paillier encryption to build a voting architecture that makes the election process transparent and tamper-proof. Ref. [14] proposed the first anonymous voting protocol for decentralized and self-counting on the blockchain, which is implemented in Ether through smart contracts and zero-knowledge proofs. Ref. [15] proposed a new framework for using blockchain in an electronic voting system using Ether and smart contracts. Ether is used to implement the Truffle framework for testing and checking smart contracts. Ref. [16] proposed a new system called a blockchain-based e-voting system. Their proposed system combines double envelope encryption and blockchain technology to present the new system. The developed system consists of three components: the voter side, the electoral commission, and the BC network. They had to use blockchain features (e.g., distributed ledger) to provide the missing availability of the Estonian e-voting system, because it is not shared for one person or node and all nodes in the system have the same copy of the latest data. In addition, the verifiability problem can be fixed using a hash function where a blockchain is used to secure the chain. The proposed system uses the same idea as the blockchain implementation, plus two interfaces (HTTP and WebSocket), the first one for controlling the nodes and the second one for P2P communication. In [17], a new framework, called a blockchain-based e-voting system, is proposed. The proposed system is a layered structure, as each layer is connected to its upper layer to ensure consistency, privacy, robustness, and security. Each node can be a human computer or a voting center, the reason is to avoid delays in each region, as the whole country enters the system, the chain is distributed on all levels, and the lowest level contains users and centers. The number of nodes is arranged in such a way as to avoid overloading the chain and to keep the speed as high as possible. The voting process is performed by having nodes vote for candidates, and eGov is responsible for identifying candidates and voters. A second, lower

level will have the data from the voting process, and here there is a blockchain to ensure the consistency and security of the data. Communication between the levels is ensured through the use of communication protocols. Ref. [18] proposed a voting system based on an Ethernet private blockchain, that uses smart contracts managed by regional nodes that verify the validity of each vote before attaching it to the blockchain. Smart contracts are used to count the votes, so that the results can be announced at the end of the election. In addition, the system allows voters to verify that their votes have been counted correctly, by using transaction IDs. Another type of small-scale election is introduced in [19], which relies on expensive homomorphic cryptographic primitives to protect voter privacy. Again, merging this resource-intensive cryptographic computation entails significant costs and limits the scheme to small-scale elections.

In [20], a new SecEVS secure e-voting system is proposed. The design of this system considers a network model and framework for e-voting systems. Regarding the network model, the authors applied it to a university campus. Ref. [21] proposed a system that uses hashes to record the voting results for each polling station, making this recording system more secure and using digital signatures to make the system more reliable. The proposed system is based on the use of blockchain technology which works on most of the same principles as in the Bitcoin system and focuses on database records. The proposed system uses permission Blockchain. A new electronic voting system called CryptoVoting was proposed by [22]. The system is based on the use of Shamir's secret sharing method and the use of blockchain technology. The proposed system tries to improve traceability and auditing regarding voting operations without using a third party. The system uses two linked blockchains, the first one for recording the voters and the voting process and the second one for counting the votes and obtaining the voting results. In addition to this, the system uses smart contracts to manage the voting process and results. The proposed system is based on blockchain technology using a multi-channel hybrid system. The main goal of the system is to allow remote voting. The system is implemented using sidechain technology. The sidechain propagates the blockchain by having new features, such as avoiding writing on the main blockchain (to reduce the cost and risk of failure and the need to create new currencies). Ref. [23] proposed a voting system based on ring signatures and blockchain, where a voter is registered and has the right to vote, and when voting they must collect all the public keys of all other voters and combine the public key with their private key according to the option chosen by the signature. Afterward, the voter uses their private key op_return, in which they transmit the encrypted ballot information. During the counting phase, the election authority collects all transactions and verifies their validity for counting purposes. The electoral authority makes the history of the transactions public, which makes it possible for the voter to verify that their vote was counted. Finally, we analyzed and summarized them, as shown in Table 2.

Blockchain-based voting protocols can solve the problem of electronic voting relying on centralized institutions, improve the anti-attack capability of the system, and increase the fairness of voting. The application of blockchain-based voting is still in the exploration stage, and further research is needed on how to balance efficiency and practicality while protecting voters' privacy; there is no widely accepted blockchain-based large-scale voting scheme in the literature. Although the use of cryptographic techniques such as Paillier encryption, proof of knowledge (PoK), and linkable ring signatures have been proposed, deploying such schemes is computationally inefficient and expensive. In addition to the informal security analysis provided in this work, voters must interact with smart contracts deployed on the blockchain to vote and validate their votes. This process requires smart contracts to perform expensive validation operations during the voting and counting phases, which can create serious performance issues as well as security and privacy concerns.

Table 2. Summary and analysis of the literature.

Research Papers	Using Technology	Safety Technology Considerations	Restrictions
[13]	BlockChain Paillier Encryption	Privacy Security	Encryption algorithms have high server requirements and cannot be used on a large scale
[14]	Blockchain Ethereum Zero Knowledge Proof Smart Contracts	Privacy Security	Lack of support for cryptography
[15]	Blockchain Ethereum Smart Contracts	Decentralization Security Authentication	Can only be applied on a small scale
[16]	Blockchain	Verifiable Privacy Valid	Dependence on centralized resources
[17]	Blockchain	Privacy Security Speed Integrity	Need to improve synchronization, latency, and performance
[18]	Go-Ethernet POA Permission Blockchain	Security Certification	They do not apply any recusal system and only review the electronic voting system
[19]	BroncoVote Homomorphic Encryption	Security Privacy	Improvements in cryptographic methods, and can only be limited to a smaller scale
[20]	Auditable Blockchain	Transparent Security Audit	Good skills are needed to apply this system and associate it with the needed nodes and agents that will have different aspects to do.
[21]	Blockchain	Reliable Safety Certification	It does not support complex applications, and the number of nodes increases their time
[22]	Crypto Voting Blockchain	Security	They do not implement any evidentiary system, they only review the electronic voting system
[23]	Blockchain Ring Signature	Privacy and Security Verifiable	Does not meet the need for fairness and no receipts

3. Anonymous Voting Protocol

To address the problems of data loss and privacy leakage caused by the single point of server failure of existing trusted third-party-based voting systems in the market, and the privacy leakage problems of existing blockchain-based voting systems; this section designs a method to insert the identity commitment generated from the user's identity information into the Merkle tree and prove the existence of the current user's identity commitment in the Merkle tree, thus ensuring the anonymity and verifiability of the system. Finally, the verification process is optimized by exploiting the advantages of on-chain and off-chain. The design and implementation of this anonymous voting scheme are presented in the following section.

3.1. Algorithm Description

The algorithmic flow of the method used in this paper is shown in Figure 1, where the band computation problem in step 1 is inserted into the Merkle tree according to the identity commitment used in this paper, then verified, the design is encoded according to the scheme, and then transformed into a low-order circuit, immediately before transforming it into a QAP (quadratic arithmetic program) problem, it still needs to be transformed into an R1CS (rank-1 constraint system) structure, to allow for easier QAP transformation and h use of zero-knowledge proof algorithms. To ensure that non-interactive zero-knowledge z-proofs are secure, a trusted setting is needed to generate a common string for generating

proofs and verifying proofs [24]. Finally, the verification process is optimized according to the characteristics of non-interactive zero-knowledge proofs. In this paper, the proofs are generated off-chain and the verification part is placed on-chain, where users generate proofs off-chain according to the proof key and then upload them to the chain, and the chain verifies them according to the deployed verification contract and the verification key.

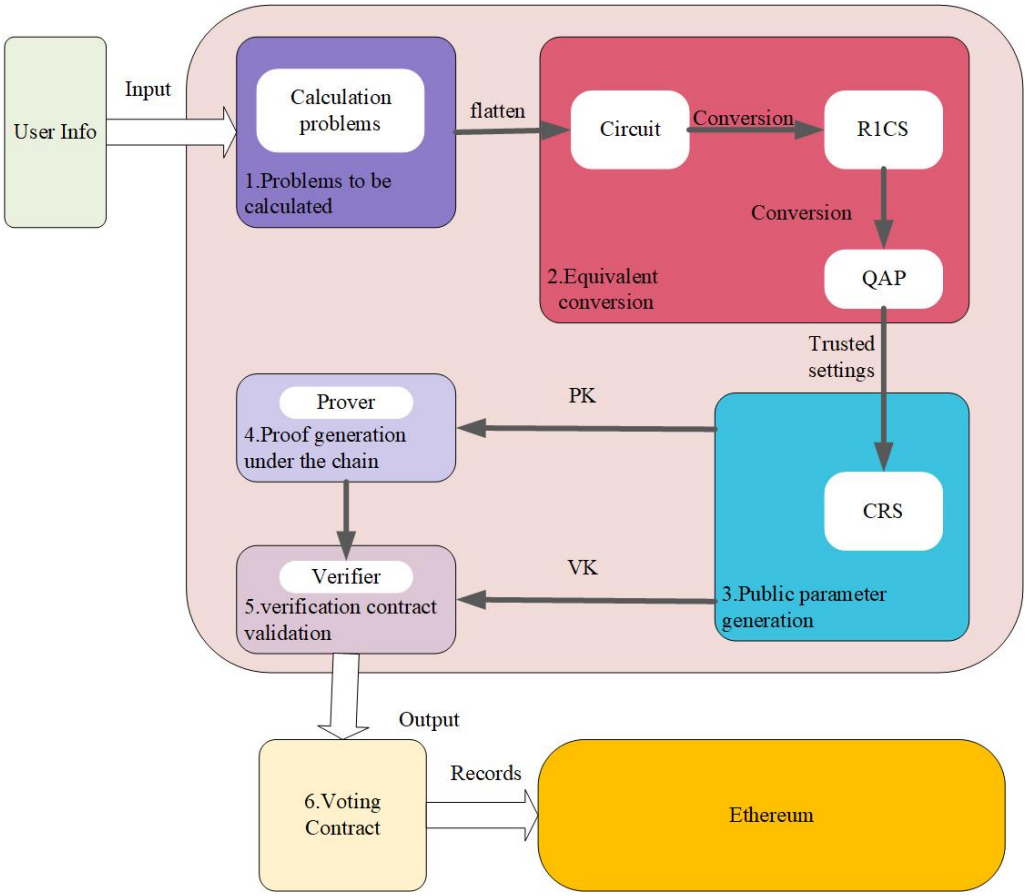

Figure 1. Algorithm schematic.

An arithmetic circuit is a directed acyclic graph over a finite field $\mathbb{F}_p$, with nodes as addition and multiplication gates and edges as connecting wires. Wires connect the output of one gate to the input of the other gate. Each gate has two input lines and one output line, and the circuit has one final output line. Figure 2 shows an example of an arithmetic circuit.

For an m-gate, n-wire circuit, define an assignment of witness $\alpha = (\alpha_1, \alpha_2, \alpha_3 ... \alpha_n)$ to the n wires of the circuit, such that the inputs and outputs of each gate satisfy the constraints defined by the gate operation. An m-gate, n-line arithmetic circuit defines a relation on the witness $\alpha = (\alpha_1, \alpha_2, \alpha_3 ... \alpha_n)$, such that some constants $u_{i,q}, v_{i,q}, w_{i,q}{}_{(1 \le i \le n, 1 \le q \le m)}$ are satisfied, Equation (1).

$$\sum_{i=1}^{n} \alpha_i u_{i,q} \cdot \sum_{i=1}^{n} \alpha_i v_{i,q} = \sum_{i=1}^{n} \alpha_i w_{i,q} \quad \forall 1 \le q \le m \tag{1}$$

The above constraint is a set of m rank-1 constraints that describe the relationship imposed by the multiplication gates of the circuit on the input and output lines. An example of a particular rank-1 constraint is S1·S2−S3 = 0, corresponding to a multiplicative gate that accepts inputs S1, S2, and output S3. A set of m rank-1 constraints can be generalized to a quadratic arithmetic program, which makes it natural for the QAP to be reduced to an arithmetic circuit.

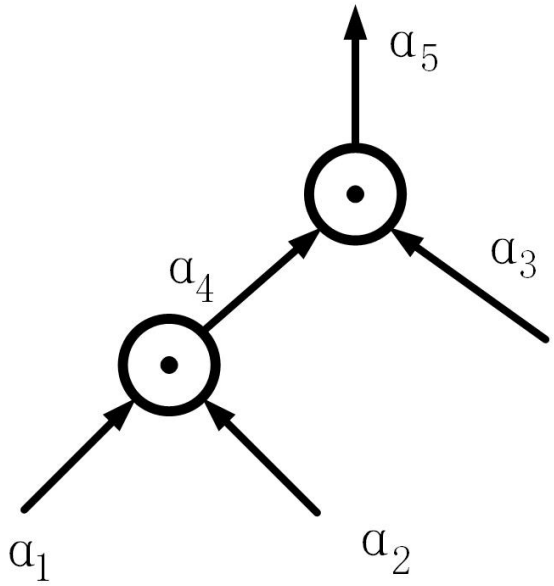

Figure 2. Example of an arithmetic circuit $f(\alpha 1, \alpha 2, \alpha 3) = ((\alpha 1 \cdot \alpha 2) \cdot \alpha 3)$.

Pick target points $r_1, r_2, ...r_m \in \mathbb{F}_p$. Define $t(x) = \prod_{q=1}^{m}(x - r_q)$. Further, let $u_i(x), v_i(x)$, $w_i(x)$ be degree m-1 polynomials, such that for $1 \leq i \leq n, 1 \leq q \leq m$.

$$\begin{aligned} u_i(r_q) &= u_{i,q} \\ v_i(r_q) &= v_{i,q} \\ w_i(r_q) &= w_{i,q} \end{aligned} \tag{2}$$

Then, a quadratic arithmetic program is a relation over $\alpha = (\alpha_1, \alpha_2, \alpha_3 ... \alpha_n)$, such that

$$\sum_{i=1}^{n} \alpha_i u_i(x) \cdot \sum_{i=1}^{n} \alpha_i v_i(x) - \sum_{i=1}^{n} \alpha_i w_i(x) \equiv 0 \quad (mod\ t(x)) \tag{3}$$

R1CS is an intermediate medium in the whole system, because the logic relations represented by the higher-order circuit are very complex and it is difficult to derive the polynomial directly. We need to compile it into a low-order circuit. Then, with the help of an intermediate medium, the low-order circuit is first converted into a set of matrix relations, as shown in Figure 3, where A/B/C are the coefficient matrices of the linear combinations of the constraint variables, respectively, and $\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_n$ are the variables that are constrained.

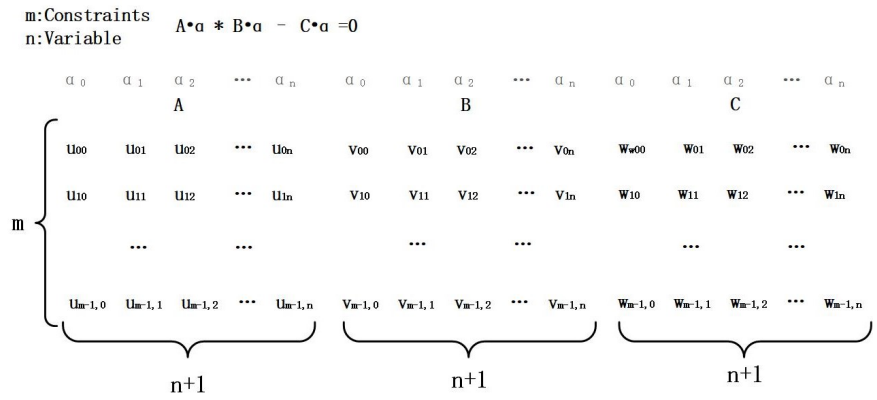

Figure 3. Description of R1CS.

To make it possible to solve the problem described by its circuit using the Groth16 algorithm in the non-interactive zero-knowledge proof, it is also necessary to speak about the conversion of RC1S into a QAP (quadratic arithmetic program) problem by Lagrange interpolation. What we have to do is to take the first value from the vector of each coefficient matrix (i.e., each column of the coefficient matrix) and use the Lagrange interpolation of Equation (4) to obtain the polynomial, and finally, we will obtain Equation $u_i(x), v_i(x), w_i(x) 1 \leq i \leq n$. Finally, under $t(x) = (x-1)(x-2)\ldots(x-n)$, compute $h(x)$, such that Equation (5) is satisfied.

$$\ell_j(x) := \prod_{i=0, i\neq j}^{k} \frac{x - x_i}{x_j - x_i} = \frac{(x-x_0)}{(x_j - x_0)} \cdots \frac{(x-x_{j-1})}{(x_j - x_{j-1})} \frac{(x-x_{j+1})}{(x_j - x_{j+1})} \cdots \frac{(x-x_k)}{(x_j - x_k)} \tag{4}$$

$$(\sum_{i=1}^{n} a_i u_i(x)) \cdot (\sum_{i=1}^{n} a_i v_i(x)) - (\sum_{i=1}^{n} a_i w_i(x)) = h(x)t(x) \tag{5}$$

The next step is to generate the proof and verify the proof, as follows.

1. Select a generator g and a cryptographic pairing e.
2. For a function $f(u) = y$ with n total variables, of which m are input/output variables, convert into the polynomial form $(u_i(x), v_i(x), w_i(x)_{i\in 0,\ldots,n}, t(x))$, of degree d (equal to the number of operations) and size $n+1$.
3. Sample random $s, \rho_u, \rho_v, \alpha_u, \alpha_v, \alpha_w, \beta, \gamma$.
4. Set $\rho_w = \rho_u \cdot \rho_v$ and the operand generators $g_u = g^{\rho_u}, g_r = g^{\rho_v}, g_w = g^{\rho_w}$.
5. Set the proving key, as in Equation (6)–(8):

$$\left(\{g^{s^k}\}_{k\in[d]}, \{g_u^{u_u(s)}, g_v^{v_i(s)}, g_w^{w_i(s)}\}_{i\in\{0,\ldots,n\}}, \right. \tag{6}$$

$$\{g_u^{\alpha_u u_i(s)}, g_v^{\alpha_v v_i(s)}, g_W^{\alpha_W w_i(s)}, g_u^{\beta u_i(s)} g_v^{\beta v_i(s)} g_w^{\beta w_i(s)}\}_{i\in(m+1,..,n)}, \tag{7}$$

$$\left. g_u^{t(s)}, g_v^{t(s)}, g_w^{t(s)}, g_u^{a_u t(s)}, g_v^{\alpha_v t(s)}, g_w^{\alpha_w t(s)}, g_u^{\beta t(s)}, g_w^{\beta t(s)}, g_w^{\beta t(s)}\right) \tag{8}$$

6. Set the verification key Equation (9):

$$(g^1, g_w^{t(s)}, \{g_u^{u_i(s)}, g_v^{v_i(s)}, g_w^{w_i(s)}\}_{i\in\{0,\ldots,m\}}, g^{\alpha_u}, g^{\alpha_v}, g^{\alpha_w}, g^{\gamma}, g^{\beta\gamma}) \tag{9}$$

7. Generate proof:
(1). For input h, perform the computation of f(h) to obtain the values of the intermediary variables $r_{ii\in m+1,\ldots,n}$.
(2). Assign all values to the unencrypted variable polynomials Formula (10), and similarly V(x), W(x).

$$U(x) = u_0(x) + \sum_{i=1}^{n} r_i \cdot u_i(x) \tag{10}$$

(3). Sample random $\delta_u, \delta_v, \delta_w$.
(4). Find $h(x)$, Formula (11).

$$h(x) = \frac{U(x)V(x) - W(x)}{t(x)} + \delta_v U(x) + \delta_u V(x) + \delta_u \delta_v t(x) - \delta_w \tag{11}$$

(5). Assign the prover's variable values to the encrypted variable polynomials and apply zero-knowledge δ-shift, Formula (12), and similarly$V(x), W(x)$.

$$g_u^{U_p(s)} = (gut(s))^{\delta_u} \cdot \prod_{i=m+1}^{n} (g_u^{u_i(s)})^{r_i} \tag{12}$$

(6). Assign its α-shifted pairs, Formula (13), and similarly $V(x), W(x)$.

$$g_u^{U'_p(s)} = (g_u^{\alpha_u t(s)})^{\delta_u} \cdot \prod_{i=m+1}^{n} (g_u^{\alpha_u u_i(s)})^{r_i} \tag{13}$$

(7). Assign the variable values consistency polynomials, Formula (14).

$$g^{Z(s)} = (g_u^{\beta t(s)})^{\delta_u} (g_v^{\beta t(s)})^{\delta_v} (g_w^{\beta t(s)})^{\delta_w} \cdot \prod_{i=m+1}^{n} (g_u^{\beta u_i(s)} g_v^{\beta v_i(s)} g_w^{\beta w_i(s)})^{r_i} \tag{14}$$

(8). Compute the proof, Formula (15).

$$(g_u^{U_p(s)}, g_v^{R_p(s)}, g_w^{W_p(s)}, g^{h(s)}, g_u^{U'_p(s)}, g_v^{V'_p(s)}, g_w^{W'_p(s)}, g^{Z(s)}) \tag{15}$$

8. Verification Proof:
(1). Parse a provided proof as Formula (16).

$$(g_u^{U_p}, g_v^{V_p}, g_w^{W_p}, g^h, g_u^{U'_p}, g_v^{V'_p}, g_w^{W'_p}, g^Z) \tag{16}$$

(2). Assign input/output values to the verifier's encrypted polynomials and add to 1: Formula (17) and similarly $V(x), W(x)$.

$$g_u^{U_r(s)} = g_u^{u_0(s)} \cdot \prod_{i=1}^{m} (g_u^{u_i(s)})^{r_i} \tag{17}$$

(3). Variable polynomials restriction check: Formula (18) and similarly $V(x), W(x)$.

$$e(g_u^{U_p}, g^{\alpha_u}) = e(g_u^{U'_p}, g) \tag{18}$$

(4). Variable values consistency check: Formula (19) [25].

$$e(g_u^{U_p} g_v^{V_p} g_w^{W_p}, g^{\beta\gamma}) = e(g^Z, g^\gamma) \tag{19}$$

3.2. Anonymous Authentication Scheme Design

In this paper, we use the Merkle tree to store the anonymous identity of voters, because the Merkle tree has the features of efficient verification, security, easy scaling, and space-saving, which is very suitable for verifying the integrity of data and protecting the security of digital assets. For example, 3 levels of Merkle can store the anonymity of 2^3, i.e., 8 voters. In this paper, we take 20 levels as an example, which can store 2^{20} voters' information, i.e., 1,048,576 voters.

In distributed anonymous voting, each voter has only one vote with uniqueness. To ensure the anonymity of the user without revealing the user's identity information, this scheme adopts the approach shown in Figure 4, we use the hash plus signature approach. The hash function has the problem of hash collision, to prevent the hash collision, we use the signature to ensure the uniqueness of the generated anonymous information, and finally the signature information is divided into two, one as a security factor and one as a unique identifier representing the user's identity. Finally, the secondary hash of the two is used as the anonymous identity of the user and stored in the tree for authentication during voting.

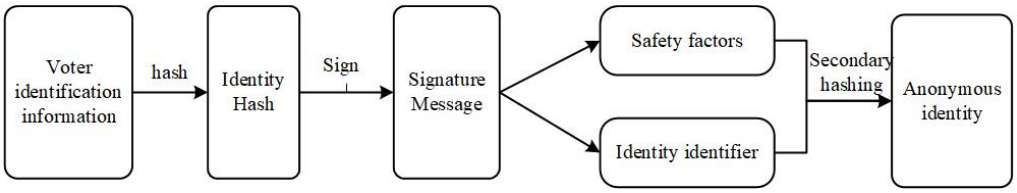

Figure 4. Voter identity encryption.

According to the algorithm described in Section 3.1, we need to design a circuit for voter identity verification, where according to the previous description [26], we take a Merkle tree to store the user's anonymous identity, then, we also need to design a verification circuit based on Merkle, as shown in Figure 5, according to the path index of the input target node to the root node, where the Merkle tree uses a binary tree. The path index represents the location information of the target node in the tree, represented by the left 0 right 1. The identity commitment generated by the sibling node and the user information in the process of generating the root node is used to determine the user's identity, and the combination of the generated identity commitment and the provided target node index and sibling node is thus used to verify the voter information. To ensure the uniqueness of the user ballot, we use the identity identifier and external identifier after hashing as the ballot credentials, and the ballot information hash is for anonymous authentication of the voter voting marquis, to see whether the vote they cast is their own, whether the voting information is tampered with, etc.

Finally, the circuit is compiled into R1CS, and then the algorithm of Section 3.1 is used to transform and prove the related information generation. In other words, it means that in the process of user voting, we only need to judge whether the information provided by the user satisfies its constraint system.

One of the hash functions used in the circuit is the Poseidon function, which is a hash algorithm specifically designed for zero-knowledge proof (ZKP) cryptographic protocols, and it excels in the computational complexity of proof generation and verification. Compared with other similar algorithms, such as SHA-256, SHA-3, and Pedersen hash functions, Poseidon can significantly improve the operational efficiency of the entire zero-knowledge proof system in scenarios where zero-knowledge proofs are applied. This algorithm is designed to protect user privacy while ensuring data integrity and security [27].

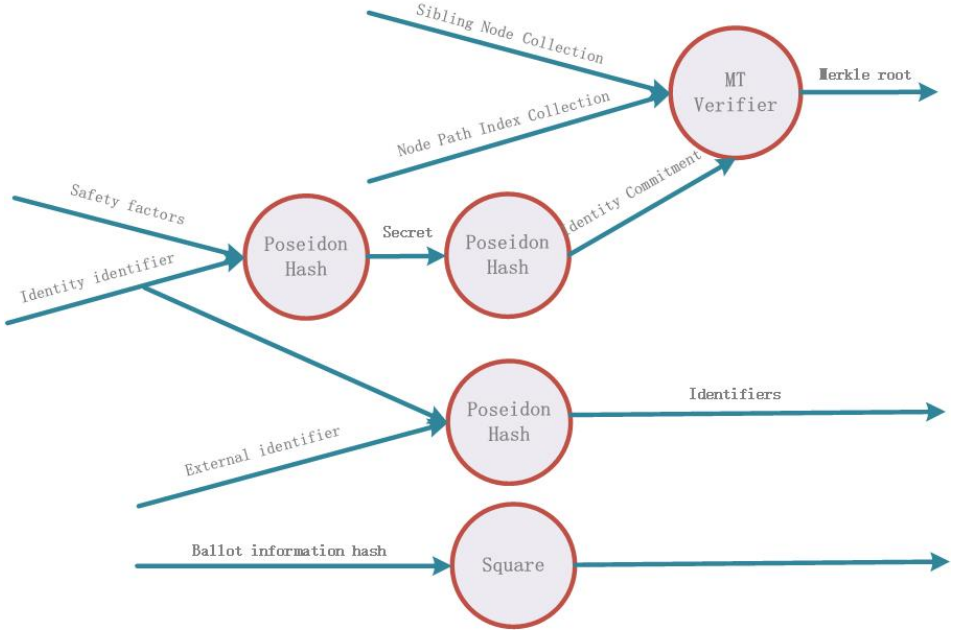

Figure 5. Verification circuit description.

3.3. Anonymous Voting Solution Implementation

This section introduces the implementation process of anonymous voting, as shown in Figure 6, where the circuit is coded in the circom language [23], and then the code is written according to the circuit design in Section 3.2, and then compiled by its circom compiler, and the required proof key and verification key are generated by the trusted settings. The proofs are generated and verified according to the inputs and outputs in the circuit diagram, as well as the proof and verification keys.

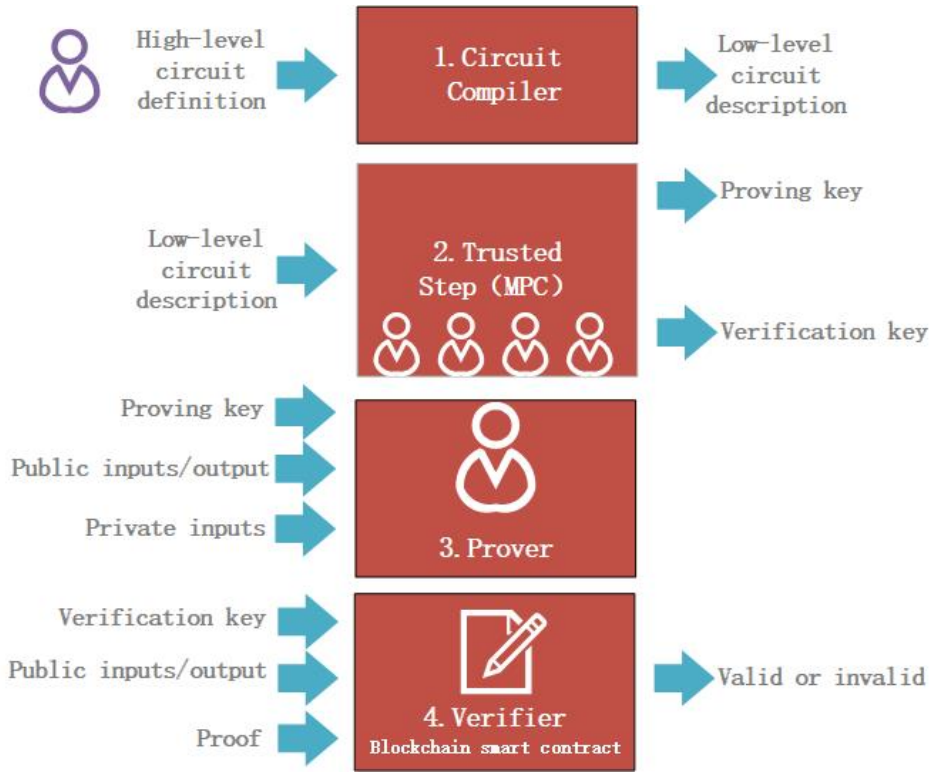

Figure 6. Implementation flow.

The circom file is created and coded to implement the circuit described in Section 3.2. Different levels are set and the system can be compiled to obtain different amounts of constraints. In this paper, we set the tree to 20 levels and the compilation can generate 5554 constraints, private input 42, common input 2, and output 2, etc., as shown in Figure 7. Four files are also generated at the end, for the generation and verification of the proofs later, as shown in Table 3.

```
PS D:\data\dVoting-master\dVoting-master\buildscircom>
[INFO]  snarkJS: Curve: bn-128
[INFO]  snarkJS: # of Wires: 5579
[INFO]  snarkJS: # of Constraints: 5554
[INFO]  snarkJS: # of Private Inputs: 42
[INFO]  snarkJS: # of Public Inputs: 2
[INFO]  snarkJS: # of Labels: 17623
[INFO]  snarkJS: # of Outputs: 2
PS D:\data\dVoting-master\dVoting-master\buildscircom>
```

Figure 7. Circuit compiled information.

Table 3. Introduction to compilation files.

File Formats	Introduction
r1cs	R1CS constraint system for circuits in binary format
wasm	Generate directories of additional files needed for witnessing
ptau	The ptau file contains a history of all challenges and responses that have occurred to date
zkey	The file containing the key pair

Now, the next trusted setting can be made. The trusted setting we use is the powers of tau ritual [28], first create a tau ritual, the type of curve used is a bn128 curve, specify how large the circuit is to generate safe parameters, in this paper the constraint is 5554, to prevent the later expansion of it we use 214 constraints, that is, for 16384 constraints to generate safe parameters. Immediately after that, some random parameters need to be provided to provide additional entropy sources, and finally, the random beacon technique is used to make its generated parameters more secure. Random beacons are sources of public randomness and are not available until a fixed time. The beacon itself can be a delayed hash function. The cryptographic evaluation of the Lagrangian polynomials for tau, alpha*tau, and beta*tau is performed using the prepare phase2 command. It takes the beacon [29] ptau file we generated in the previous step and outputs a final ptau file that will be used to generate the circuit proof and verify the key. Finally, the generated file is verified and the verification passes to proceed to the next step, as shown in Figure 8, where the verification process prints out the challenge and the corresponding hash for each one.

After creating the specific powers of tau in the previous step, we need to set up our design circuit to generate the proof and verification keys we need. First, we need to set up the compiled r1cs file of the circuit according to the Groth16 algorithm and the final ptau file, computed in the previous step, to generate the required verification and proof keys; the result of this stage cannot be used directly to generate a key pair, which requires at least two rounds of contributions and the setting of random beaconing techniques to add additional sources of entropy, thereby achieving a more secure key pair; and finally, it will generate a zkey file, which contains the proof key and verification key required for the zero-knowledge proof. Next, we verify that the key file is available based on the constraint file and the multiparty ceremony, and finally, the verification passes to be usable.

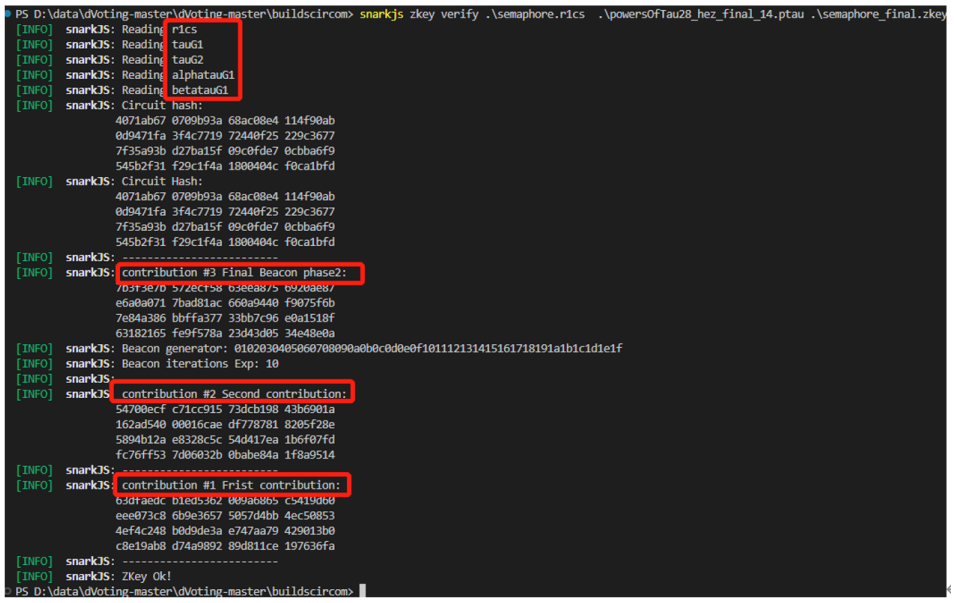

Figure 8. Circuit compiled information.

To relieve the pressure of on-chain computation, we take advantage of the combination of on-chain and off-chain to improve the efficiency of the system. We deploy the verification key and verification code to the blockchain through a smart contract language for voter identity verification during the voting process.

The user identity verification in the user voting process is performed to generate the security factor and identity identifier, and then both are taken to hash to obtain the identity commitment, as shown in Figure 9. Then, the information on the tree where the user's current user is located is obtained, as shown in Figure 10, which contains the current node, the path from the current node to the root node, the root node, and the sibling nodes in the process of generating the root node.

Based on the information obtained above, the circuit is compiled to form a wasm file and a ptau file containing the proof key, and the proof is generated by the Groth16 algorithm. The proof and additional identifier, along with the identity identifier, identity pledge, and ballot information, are then uploaded to the chain for verification, since the verification contract already contains the proof key and verification algorithm. As long as the uploaded information is verified, the identity of the user can be determined anonymously, without revealing additional information about the user.

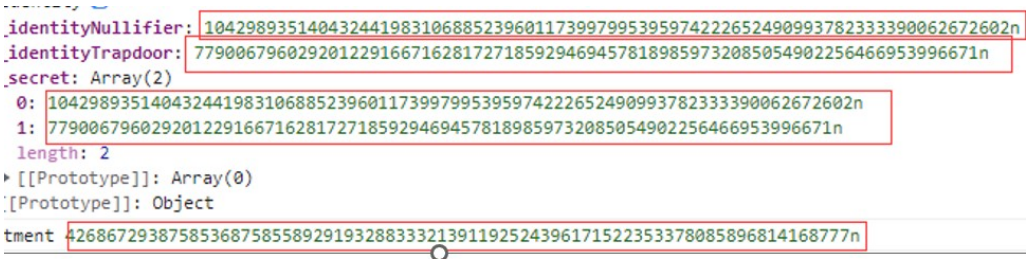

Figure 9. Circuit compiled information.



Figure 10. Circuit compiled information.

In terms of voting uniqueness, we use the identity identifier and external identifier hash as the criteria to determine whether to vote or not, if the user votes for the first time,

they can vote smoothly, otherwise, their identifier will be recorded on the blockchain, and all users will check whether their identifier is used before voting, and those whose identifier has been used will not be able to vote, as shown in Figure 11, and the second use of the same identifier will result in an error.

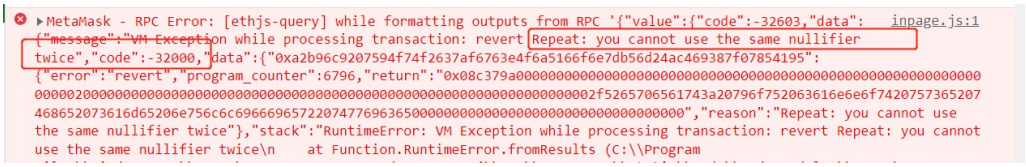

Figure 11. Circuit compiled information.

In this paper, we have tested the performance of this method through Google Chrome, and the time spent from the start of voting to the final successful vote is 0 to 1.8 min, the detailed time analysis is shown in Figure 12. From the figure, we can see that the time that accounts for the largest proportion is the response time of calling third-party plug-ins to pay for the fees required to execute the contract on Ether, and that after removing the loading, rendering, drawing, and system execution, etc., the time to execute the script, i.e., the generation of proofs and invoking contract validation, etc., is only 10,109 ms, which is a good overall performance and can be deployed on both licensed and unlicensed Ethereum blockchains.

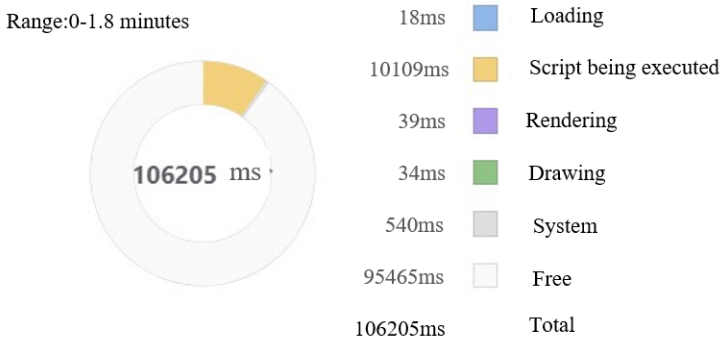

Figure 12. Voting process performance analysis.

4. Security Analysis

In this paper, we propose a smart contract-based distributed anonymous voting method, that features anonymous voting and automatic vote counting by smart contracts [30]. The security of this scheme, based on the Ethernet-distributed anonymous voting system, can achieve the features of a publicly verifiable and tamper-proof voting process. The anonymous authentication scheme proposed in this section satisfies the six characteristics of privacy, integrity, legality, fairness, completeness, and irreproducibility that its voting method has, as well as some other characteristics, as follows:

(1) Privacy: On the one hand, there is the inherent anonymity of the Ether technology, which uses information that is not related to the real identity information for voting, and uses hash functions and digital signatures to encrypt the transmitted data information, to prevent third parties from snooping . On the other hand, this scheme has a perfect protection mechanism for the voter's identity. Voters first have to use their real identity information for qualification and identity verification in the qualification stage, and after passing it, they will get an anonymous identity that is unrelated to their real identity, and then they only need to consider whether the anonymous identity is legal, without considering whether the voter's real identity is qualified to vote. Therefore, this scheme can well protect the privacy of voters.

(2) Legitimacy: This scheme can ensure the legitimacy of voters to a large extent, by conducting voter eligibility and identity verification, while third-party voters who are

not qualified to vote cannot pass the eligibility and identity verification and thus cannot register and vote. In addition, it is difficult for third parties to falsify their identities to vote, mainly for two reasons: one is that in the qualification stage, if the identity information is not true, the registration cannot be registered and directly called back, and the other is that the method uses multiparty computing technology to generate the required parameters for zero-knowledge proof, which proves difficult to falsify, and each anonymous identity can only be used once, reducing the possibility of illegitimate third parties accessing the system.

(3) Irreproducibility: This solution can prevent unlawful acts, such as multiple voting by voters, without revealing their true identities. On the one hand, a legitimate voter can only pass the administrator's review at most once, and the anonymous identity generated after the review is passed cannot be reviewed to generate a new anonymous identity. On the other hand, the voter's identifier will be recorded in the blockchain after the voting is completed, to prevent the secondary use of the anonymous identity.

(4) Integrity: This solution ensures that the votes of all legitimate voters are correctly counted by the counting contract, as voters can verify the correctness of the counting results of the counting contract. This verification includes both personal verifiability and global verifiability. In particular, individual verifiability means that voters can verify that their votes are correctly counted, while global verifiability ensures that all legitimate votes are counted by the counting contract, i.e., all legitimate votes are counted in the count. This validation mechanism can effectively prevent cheating by the counting organization and ensure the fairness and transparency of the voting.

(5) Impartiality: In the vote counting stage, only after the administrator has voted within the specified time and the administrator has finished the election, will the system invoke the smart contract to call the counting contract to count the vote results. Therefore, as long as there is a vote made after the administrator has finished the election, the vote results cannot be counted. Therefore, the program is fair.

(6) Secure and reliable information storage: In this paper, we use the Merkle tree to store the anonymous identity information of voters, which can well ensure the integrity of user information, and any tampering of information will lead to hash inconsistency and thus be detected. The security is high.

(7) Proof parameter security: The zk-SNARK [31] algorithm requires a trusted initial public parameter, and if this parameter is tampered with by an attacker or by an untrustworthy third party, then the whole system will have security problems. In this scheme, we introduce the powers of tau method. The method is generated based on mathematical principles, which can guarantee that the generated parameters are secure and trustworthy and will not be tampered with or attacked, thus improving the security of the system to a great extent. The generation and verification of proofs during the user–contract interaction in this scheme are calculated based on the above circuit, which can guarantee the voter's identity information to a great extent.

(8) Uncontestedness: This scheme uses blockchain as a channel for identity verification of voters' Ether accounts and voter information, and any invalid account cannot vote in the voting system. The method also limits one person to one vote without revealing voter information, and each ballot can only select one candidate, ensuring that the voting results are accurate and the voting scheme is non-controversial.

(9) Verifiability: After voters submit their ballots, if the system has not yet finished the election, users can check whether the ballot results have been changed in the client. Because the data on the Ethernet satisfies public verifiability, voters can verify whether their ballots satisfy this circuit-bounded system by using zero-knowledge proofs based on the input of the circuit description. If the verification is successful, it returns 1, otherwise, it returns 0. Therefore, this scheme has public verifiability.

(10) Robustness: The design principle of this scheme is to ensure that all voters can vote within the specified time before the smart contract counts the votes. Even if some voters abstain from voting or exceed the voting time, the system treats them as abstentions

and counts them in the final result. Thus, the system can function properly and produce valid votes even if some voters do not actively participate in the voting process.

5. Conclusions

With the birth of Ether and the implementation of smart contract technology, blockchain technology is no longer limited to the financial environment but can be applied to various fields, such as the energy industry, medical and health care, internet of things, retail and logistics, government and public services, etc. In this paper, we design and develop a distributed anonymous e-voting system, based on Ether, that solves the problems of a single point of failure, data loss, and leakage brought by traditional methods, while the hashing and linking of blocks in Ether make the voting data mathematically unmodifiable or unchangeable, solving the problem of tampering with ballots in traditional e-voting. Finally, based on zero-knowledge proof and Merkle technology, and with the advantage of combining on-chain and off-chain, it solves the security problem when users interact with the contract.

Author Contributions: Conceptualization, W.T. and W.Y.; methodology, W.T. and X.T.; validation, W.T.; writing—original draft preparation, W.T. and S.Y.; writing—review and editing, W.T., W.Y. and X.T.; supervision, W.T. and W.Y.; funding acquisition, W.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Science Foundation of China, grant number 202204120017; the Autonomous Region Science and Technology Program, grant number 2022B01008-2; and Autonomous Region Science and Technology Program, grant number 2020A02001-1.

Data Availability Statement: No data was used for the research described in the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Taş, R.; Tanrıöver, Ö. A Manipulation Prevention Model for Blockchain-Based E-Voting Systems. *Secur. Commun. Netw.* **2021**, *2021*, 6673691. [CrossRef]
2. Yi, H. Securing e-voting based on blockchain in P2P network. *Eurasip J. Wirel. Commun. Netw.* **2019**, *2019*, 137. [CrossRef]
3. Fan, W.; Kumar, S.; Jadhav, V.; Chang, S.Y.; Park, Y. A Privacy Preserving E-Voting System Based on Blockchain. In Proceedings of the Silicon Valley Cybersecurity Conference, San Jose, CA, USA, 17–19 December 2020; Springer: Cham, Switzerland, 2020; pp. 148–159.
4. Halderman J.A.; Teague, V. The New South Wales iVote system: Security failures and verification flaws in a live online election. In Proceedings of the International Conference on e-Voting and Identity, Bern, Switzerland, 2–4 September 2015; Springer, Cham, Switzerland, 2015; pp. 35–53.
5. Springall, D.; Finkenauer, T.; Durumeric, Z.; Kitcat, J.; Hursti, H.; MacAlpine, M.; Halderman, J.A. Security analysis of the Estonian internet voting system. In Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, AZ, USA, 3–7 November 2014; pp. 703–715.
6. Nakamoto, S.; Bitcoin, A. A Peer-to-Peer Electronic Cash System. *Bitcoin* **2008**, *4*. Available online: https://bitcoin.org/bitcoin.pdf (accessed on 9 March 2023).
7. Song, J.G.; Moon, S.J.; Jang, J.W. A scalable implementation of anonymous voting over Ethereum blockchain. *Sensors* **2021**, *21*, 3958. [CrossRef] [PubMed]
8. Wang, H.; Wang, Y.; Cao, Z.; Li, Z.; Xiong, G. An overview of blockchain security analysis. In Proceedings of the China Cyber Security Annual Conference, Beijing, China, 14–16 August 2018; Springer: Singapore, 2018; pp. 55–72.
9. Shinde, S.S.; Shukla, S.; Chitre, D.K. Secure E-voting using homomorphic technology. *Int. J. Emerg. Technol. Adv. Eng.* **2013**, *3*, 203–206.
10. Shukla, S.; Thasmiya, A.N.; Shashank, D.O.; Mamatha, H.R. Online voting application using ethereum blockchain. In Proceedings of the 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Bangalore, India, 19–22 September 2018; pp. 873–880.
11. Al-Maaitah, S.; Qatawneh, M.; Quzmar, A. E-Voting System Based on Blockchain Technology: A Survey. In Proceedings of the 2021 International Conference on Information Technology (ICIT), Guangzhou, China, 22–25 December 2021; pp. 200–205.
12. Yu, B.; Liu, J.K.; Sakzad, A.; Nepal, S.; Steinfeld, R.; Rimba, P.; Au, M.H. Platform-Independent Secure Blockchain-Based Voting System. In Proceedings of the 21st International Conference, ISC 2018, Guildford, UK, 9–12 September 2018.
13. Gupta, S.P.; Tripathi, A.M. E-Voting using Blockchain. *J. Phys. Conf. Ser.* **2021**, *1911*, 012001. [CrossRef]

14. McCorry, P.; Shahandashti, S.F.; Hao, F. A smart contract for boardroom voting with maximum voter privacy. In *Financial Cryptography and Data Security, Proceedings of the International Conference on Financial Cryptography and Data Security, Sliema, Malta, 3–7 April 2017*; Springer: Cham, Switzerland, 2017; pp. 357–375.
15. Patidar, K.; Jain, S. Decentralized e-voting portal using blockchain. In Proceedings of the 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, 6–8 July 2019; pp. 1–4.
16. Adiputra, C.K.; Hjort, R.; Sato, H. A proposal of blockchain-based electronic voting system. In Proceedings of the 2018 Second World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4), London, UK, 30–31 October 2018; pp. 22–27.
17. Bulut, R.; Kantarcı, A.; Keskin, S.; Bahtiyar, Ş. Blockchain-based electronic voting system for elections in Turkey. In Proceedings of the 2019 4th International Conference on Computer Science and Engineering (UBMK), Samsun, Turkey, 11–15 September 2019; pp. 183–188.
18. Hjálmarsson, F.Þ; Hreiðarsson, G.K.; Hamdaqa, M.; Hjálmtýsson, G. Blockchain-based e-voting system. In Proceedings of the 2018 IEEE 11th International Conference on Cloud Computing (CLOUD), San Francisco, CA, USA, 2–7 July 2018; pp. 983–986.
19. Dagher, G.G.; Marella, P.B.; Milojkovic, M.; Mohler, J. BroncoVote: Secure Voting System using Ethereum's Blockchain. In Proceedings of the 4th International Conference on Information Systems Security and Privacy, Funchal, Portugal, 22–24 January 2018.
20. Pawlak, M.; Poniszewska-Marańda, A.; Kryvinska, N. Towards the intelligent agents for blockchain e-voting system. *Procedia Comput. Sci.* **2018**, *141*, 239–246. [CrossRef]
21. Hanifatunnisa, R.; Rahardjo, B. Blockchain based e-voting recording system design. In Proceedings of the 2017 11th International Conference on Telecommunication Systems Services and Applications (TSSA), Lombok, Indonesia, 26–27 October 2017; pp. 1–6.
22. Fusco, F.; Lunesu, M.I.; Pani, F.E.; Pinna, A. Crypto-voting, a Blockchain based e-Voting System. In Proceedings of the KMIS, Seville, Spain, 18–20 September 2018; pp. 221–225.
23. Wu, Y. An e-Voting System Based on Blockchain and Ring Signature. Master's Thesis, University of Birmingham, Birmingham, UK, 2017.
24. Chen, T.; Lu, H.; Kunpittaya, T.; Luo, A. A Review of zk-SNARKs. *arXiv* **2022**, arXiv:2202.06877.
25. Petkus, M. Why and how zk-snark works: Definitive explanation. *arXiv* **2019**, arXiv:1906.07221.
26. Bellés-Muñoz, M.; Baylina, J.; Daza, V.; Munoz-Tapia, J.L. New privacy practices for blockchain software. *IEEE Softw.* **2021**, *39*, 43–49. [CrossRef]
27. Grassi, L.; Khovratovich, D.; Rechberger, C.; Roy, A.; Schofnegger, M. Poseidon: A New Hash Function for Zero-Knowledge Proof Systems. In Proceedings of the USENIX Security Symposium, Virtual event, 11–13 August 2021.
28. Koh WeiJie. Available online: https://medium.com/coinmonks/announcing-the-perpetual-powers-of-tau-ceremony-to-benefit-all-zk-snark-projects-c3da86af8377 (accessed on 16 March 2023).
29. Bowe, S.; Gabizon, A.; Miers, I. Scalable Multi-Party Computation for zk-SNARK Parameters in the Random Beacon Model. *Cryptology ePrint Archive*, 2017. Available online: https://eprint.iacr.org/2017/1050 (accessed on 16 March 2023).
30. Kshetri, N.; Voas, J. Blockchain-enabled e-voting. *IEEE Softw.* **2018**, *35*, 95–99. [CrossRef]
31. Banerjee, A. A Fully Anonymous e-Voting Protocol Employing Universal Zk-SNARKs and Smart Contracts. In Proceedings of the International Congress on Blockchain and Applications, Salamanca, Spain, 6–8 October 2021; Springer: Cham, Switzerland, 2021; pp. 349–354.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.