Novel Vote Scheme for Decision-Making Feedback Based on Blockchain in Internet of Vehicles

Yongjun Ren, *Member, IEEE*, Fujian Zhu, Jin Wang, *Senior Member, IEEE*, Pradip Kumar Sharma, *Senior Member, IEEE*, and Uttam Ghosh, *Senior Member, IEEE*

***Abstract*— Obtaining timely and accurate traffic information is one of the most important problems in intelligent transportation system, which will make vehicles run smoothly, avoid road congestion, save road running time and reduce vehicle energy consumption. In the current Internet of Vehicles system, the traffic management center can learn from the feedback information of all vehicles to improve the ability of decision-making and traffic command. However, the existing feedback mechanism does not respond to the spatial-temporal characteristics of data in time, due to the lack of communication capability of the current equipment. So, it cannot meet the requirements of ultra-low delay, high reliability and high security in the Internet of Vehicles. To solve this problem, this paper proposes a blockchain-based proxy vote and revocation scheme for decision feedback in Internet of Vehicles, which allows the intelligent system to ignore the unevenness and heterogeneity in the 6G technology. In addition, blockchain technology notarizes the vote data of vehicles and outsources microservices. Secondly, we use the attributes of decision-related nodes instead of their identities to enable anonymous vote. Smart contracts can automatically expand the scalability of outsourced microservices. Finally, the security proof of the proposed scheme ensures the security and consistency of outsourced microservices. The simulation results also show that our scheme greatly improves the efficiency of voting feedback.**

***Index Terms*— Internet of Vehicles, vote feedback, blockchain, attribute-based encryption, outsourcing computation.**

I. Introduction

IN THE Internet of Vehicles (IoVs), vehicles equipped with intelligent sensing devices can obtain information such as vehicle speed, acceleration, distance from surrounding vehicles, overtaking warning and so on [1]. In the process of driving, through Vehicle to Vehicle (V2V) [2] communication and Vehicle to Infrastructure (V2I) [3] communication, the perceived information is transmitted to the surrounding vehicles, roadside units and vehicle management center, so as to realize the interconnection between vehicles, and further support various smart transportation applications, such as accident warning, road broadcasting, advertising push and resource sharing, etc., to improve the safety, comfort and convenience of driving.

The complex and changeable traffic conditions and the unpredictable driving behavior of vehicles will lead to the intermittent connection of V2V communication, the highly dynamic network topology and the unstable communication link, which affect the data transmission performance of the IoVs. Some researchers improve data forwarding by deploying roadside infrastructure to provide more powerful communication and data processing capabilities than on-board devices; some scholars use vehicles parked on the roadside to assist in data forwarding. In addition, more and more new vehicles are added every year, which makes the transportation system more complex and changeable. Thus, a large number of Artificial Intelligence (AI) technologies have been adopted to improve the intelligence of the IoVs [4].

The current smart IoV is a fusion of traditional IoVs and AI. Vehicles and smart devices on the road can continuously generate massive amounts of data and send to traffic management center to learn and perform intelligent processing [5]. The processor of vehicles and smart devices on the road has limited computing power and cannot perform system-level intelligent calculation of the traffic management center. In the future 6G network with higher bandwidth and lower latency, the amount of node data in the intelligent transportation system will also increase rapidly, but the rapid mobility of vehicles, the uneven distribution of transportation equipment, and the different speeds of network communication equipment will inevitably lead to more uneven distribution of the IoVs data in time and space. Massive heterogeneous data will increase the difficulty of data management, especially the feedback data of real-time prediction will reduce the accuracy of prediction due to uneven distribution in time and space. Therefore, the decision from the IoVs management center should pay more attention to the real-time feedback from related nodes to generate more timely and accurate decisions.

The heterogeneity of IoVs data, the complexity of the network, the wide difference between IoVs systems, and the weakness of privacy will inevitably affect the development of interconnected intelligence. So, increasing the versatility of

Manuscript received February 28, 2021; revised May 23, 2021; accepted July 9, 2021. Date of publication August 4, 2021; date of current version February 2, 2022. This work was supported by the National Natural Science Foundation of China under Grant 61772280, Grant 61772454, Grant 62072056, and Grant 62072249. The Associate Editor for this article was S. Mumtaz. *(Corresponding author: Jin Wang.)*

Yongjun Ren and Fujian Zhu are with the School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China (e-mail: renyj100@126.com; zhufujian1995@gmail.com).

Jin Wang is with the School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha 410004, China (e-mail: jinwang@csust.edu.cn).

Pradip Kumar Sharma is with the Department of Computing Science, University of Aberdeen, Aberdeen AB24 3FX, U.K. (e-mail: pradip.sharma@abdn.ac.uk).

Uttam Ghosh is with the Department of EECS, Vanderbilt University, Nashville, TN 37235 USA (e-mail: ghosh.uttam@ieee.org).

Digital Object Identifier 10.1109/TITS.2021.3100103

IoVs data can accelerate the learning progress of interconnected intelligence. Blockchain is a perfect supplement for smart IoVs [6]. It can be used to improve the interoperability, security, privacy, reliability of the existing IoVs systems. Blockchain and IoVs also have similarities. How to effectively integrate the blockchain and the IoVs is also a major challenge.

To solve the spatio-temporal data isolation of the existing learning computation in smart IoVs from the real IoVs and data heterogeneity caused by structural differences among IoVs systems. This paper proposes a simplified and general feedback scheme for smart IoVs, which uses anonymous vote to feedback the decision made by transport management center to help the management center improve the learning strategy and then make better decisions. Our major contributions in this article are summarized as the following aspects.

1) We use Attribute-Based Encryption (ABE) to build an anonymous vote scheme so that vehicles and transport infrastructures with decision-related attributes can feedback decisions by voting anonymously. The attribute limitation of vote counting ensures that only votes from the decision-related vehicles and infrastructures are effective. Using attribute collection instead of identity to mark IoVs data can simplify and generalize the vote feedback process, achieve vote goals faster.
2) To address the computation and storage resource constraints of vehicles and transport infrastructures, we allocate the most computation-intensive activities to outsourcing services and leave the rest work to vehicles and infrastructures. Similarly, vehicles and infrastructures data will be encrypted and sent to the cloud storage services to reduce their storage burden.
3) To address the problem of vehicles not always staying online, we adopted the method of proxy vote. Vehicles can delegate vote rights to trusted online infrastructures before going offline and revoke the delegation after going online. The results of the simulation experiment show that the partially online system with proxy vote spends as much time to reach vote goals as the system where all vehicles are online, and the system without proxy vote takes the same time as it takes all vehicles to go online once.
4) The hash of vote data and validation index are stored on the Blockchain [7]. The tamper-proof characteristics of Blockchain guarantees the integrity of the vote data. The correctness of the outsourcing service can be verified by checking the difference between the outsourcing data and the validation data on the chain. Moreover, smart contract can directly reward and punish outsourcing services based on the verification results.

The article is structured as follows. Section II introduces the related background and work. Section III introduces our model in detail. In Section IV, we introduce the anonymous vote scheme and the corresponding security reduction. In Section V, we show some comparative experiments on time cost between our scheme and existing conventional solutions. At last, our work is concluded in Section VI.

II. Related Works

A. Smart IoVs and Feedback

AI makes the smart IoVs more widely available. The application of AI technology in autopilot driving, auxiliary driving, voice navigation, navigation prediction, etc., has greatly promoted the development of smart IoVs. AI technology can also reduce the driving burden of drivers and improve the safety and reliability of vehicles [8]. The various novel applications of IoVs generate enormous data. After being processed by AI technology, these data can excavate the potential great value and help the traffic management center to make more efficient and reasonable decisions. The existing smart IoVs has the following characteristics.

1) Vehicles and transport infrastructures participate in deep learning calculations so that nodes do not need to upload sensor data to public servers, which can maximize the security of data privacy.
2) Vehicles and transport infrastructures use their limited computation power and data to generate narrow knowledge and models. It is necessary to further learn and calculate multiple narrow knowledge and models to obtain useful information.
3) Limiting factors such as computation power, communication capacity, the amount of data and network latency determine the learning speed and accuracy of the general model.

In the existing smart IoVs, a single smart car can only provide little smart services, and the scope is limited to a single vehicle. The overall smart IoVs system includes a large number of smart vehicles and transport infrastructures. Each vehicle and transportation infrastructure has limited intelligent computing. They collect corresponding data and send it to the transportation system management center for decision-making. The deep learning techniques commonly used in the management centers are as follows:

1) Federated Learning: computing is distributed on different types of devices, and a single machine learning model is built based on the data with different characteristics [9]. Finally, a central device combines these partial models to form the general model.

2) Knowledge Transfer Learning: in the case of the same device type, a single node uses its own limited computing power, storage capacity, and sensor data to compute knowledge and model which will be transferred to the next node in related fields [10]. The next node uses the previous knowledge and model to complete or improve its own learning task.

The idea of feedback is very common in deep learning, and the backpropagation (BP) algorithm is a widely used feedback algorithm [11]. The BP algorithm is shown in Fig. 1, which is mainly composed of three parts: input layer, hidden layer and output layer. After repeated error feedback and weight adjustment, the model function gradually approaches the real function. In the structure of the BP network, it can clearly see that repeated feedback brings the result closer to the truth.

B. Anonymous Voting

Voting is one of the effective combination strategies in ensemble learning [12]. Ensemble learning uses combination

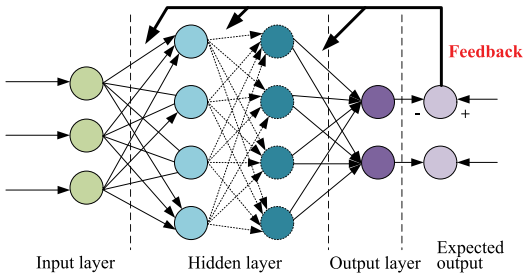


Fig. 1. The structure of BP network.

strategies to combine multiple learning methods to improve the effect of machine learning like random forest [13]. Although voting combination strategy seems very simple, it makes random forest become one of the most powerful machine learning algorithms. Samet *et al.* [14] proposed HoughNet in ECCV2020, which used a voting mechanism in the task of object detection with a top-down perspective. Thence, HoughNet can integrate short-range and long-range class-conditional evidence for visual identity and enhance the only object detection method based on local evidence.

Compared with public voting, anonymous voting can protect the privacy of nodes and ensure the independence of nodes' vote, which makes the final voting result fairer and more effective [15]. ABE can use attribute sets instead of identity to achieve anonymity [16], [17]. It can also provide secure outsourced computing functions. Li *et al.* [18] proposed an ABE scheme supporting both key generation and decryption outsourcing in 2014. This scheme can achieve the goal of outsourcing computation and verify the correctness of the results returned by the outsourcing key generation providers.

C. Blockchain for IoVs

Blockchain technology has developed rapidly. Various industries, such as manufacturing, food, and aviation, etc., have applied blockchain to their technology [19], so blockchain is leading a new round of innovation.

Blockchain was first proposed by Satoshi Nakamoto in the Bitcoin white paper published in 2008. The Ethereum was launched in 2015 and is a turing complete blockchain. The Ethereum was born with smart contract. The statement of each contract will be written into the transaction to achieve traceability and immutability [20]. Smart contracts can execute the agreement honestly and automatically.

Blockchain is a distributed ledger constructed on the Peer to Peer (P2P) network with a consensus mechanism between P2P nodes [21]. And all vehicles can be regarded as distributed nodes in the smart IoVs system. This similarity obviously provides a basis for the technology fusion of these two technologies. Blockchain for IoVs can improve the interoperability, traceability, reliability, scalability, etc., of the IoVs, making the Vehicle to Everything (V2X) as a whole [22]. Multidimensional data processed by AI will produce more specific and valuable models. Therefore, in the future, the fusion of IoVs, Blockchain and AI will be very promising.

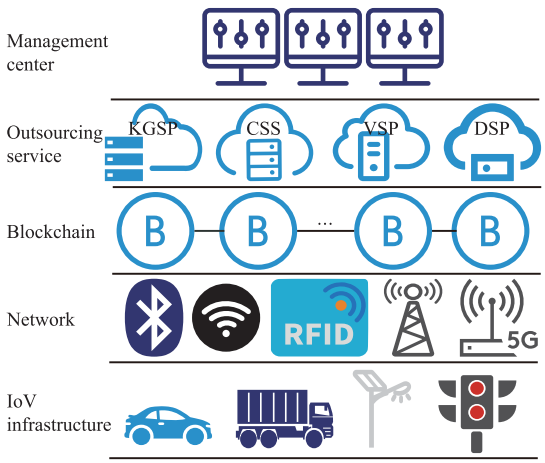


Fig. 2. System model for smart IoVs.

III. System Model

The system model of this article is shown in Fig. 2, the system model proposed in this paper is mainly divided into five layers, infrastructure layer, network communication layer, blockchain layer, oursourcing layer and management center layer. The specific introduction is as follows:

A. System Model for Smart IoVs

IoVs Infrastructure: is the foundation of the system, which collects data and feedback information. It includes mobile nodes such as smart cars, driverless vehicle, and smart trucks, as well as stable nodes such as transportation facilities, streetlights, and traffic lights.

Network Communication: is used to transmit data at the infrastructure layer and includes various methods such as Bluetooth, WiFi, RFID, base station, AP access point.

Blockchain for IoVs: provides blockchain services for the system, contains blockchain and smart contracts. Encrypted data indexes are stored on the blockchain. Smart contracts honestly and automatically verify outsourced work.

Outsourcing Service: provides computing and storage services for the IoVs system, including key generation service providers (KGSP), cloud storage service (CSS), decryption service providers (DSP), voting service providers (VSP).

Management Center: extracts the node data and feedback in each system in real time, and uses big data, artificial intelligence to process, and pushes the prediction results and decisions to each node for execution.

Based on the system model, we proposed a voting feedback model for smart IoVs based on blockchian. In Fig. 3, the specific modules and workflow are introduced below.

1) IoVs Module: the IoVs nodes send their attribute set to the attribute authority (AA) to apply the encrypted public key (PK) and then encrypt the vote data. The nodes upload the encrypted vote data to the cloud storage and send encrypted data hash, plaintext hash and PK code to the blockchain, which also completes the vote. Each IoVs node has the right to vote. When the nodes are online, they can vote for themselves, and they can also entrust the vote rights before being offline. The authorized key (AK) and PK code are sent to the host nodes

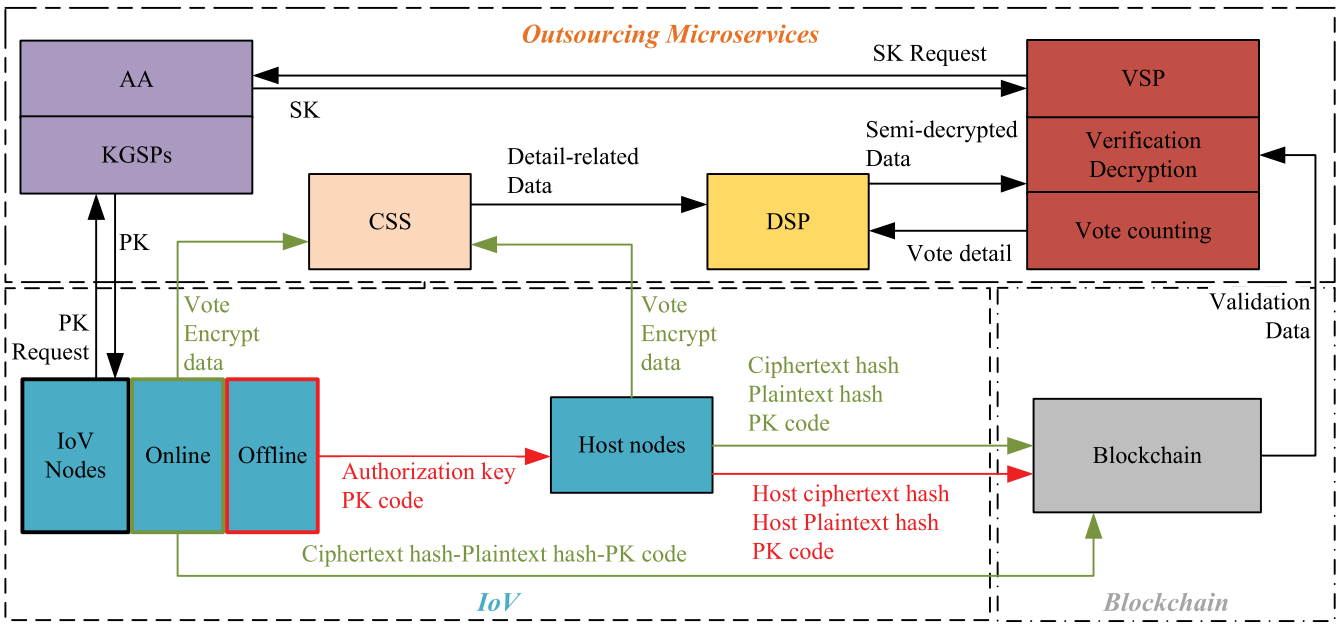


Fig. 3. Voting feedback model for smart IoVs based on blockchain.

before the nodes go offline, and the host nodes will vote on behalf of themselves and the client nodes.

2) Blockchain Module: blockchain data is open and transparent, so plaintext data cannot be uploaded to the blockchain. Moreover, the blockchain cannot directly store the encrypted IoVs data, which will increase the storage pressure and the burden of data synchronization of the blockchain nodes. Instead, the blockchain should store the index of the verification and encrypted data. The verification index is composed of the encrypted data hash, plaintext hash and code of the corresponding encrypted PK. Among them, the encrypted data hash and the PK code are used to verify the correctness of cloud storage service. The plaintext hash and the PK code are used to verify the correctness of the DSP service. The blockchain has the role of notarization because the data on the blockchain is authentic and immutable.

3) Outsourcing Module: this module fully takes over the activities which need mass calculation and storage in the system, thereby reducing the burden on IoVs nodes. Services such as key generation, storage, decryption, and vote counting all need to serve multiple nodes and multiple voting topics, so we abstract these outsourcing services to create microservices for dedicated services to achieve service stability, availability and scalability of service scale. The introduction of the service is as follows.

Key Generation Service Providers: AA is a third-party agency that is not completely trusted and will try to guess the identity of the node with the received attributes information. But AA cannot generate a key pair, the key generation is outsourced to KGSPs. KGSPs can quickly calculate many key pairs, and AA only needs to verify the correctness of the generated key pairs, which reduces the workload of AA.

Cloud Storage Service: CSS is also an incompletely trust third-party data storage service provider that can honestly execute the data storage protocol and store the encrypted voting data for IoVs nodes, but it will try to read the stored data.

Decryption Service Providers: DSP obtains the vote subject detail including attribute sets and outsourcing decryption keys from the VSP. Then DSP applies to the CSS for the vote subject-related encrypted data, decrypts and sends the semi-decrypted data to VSP. DSP undertakes massive decryption calculations and obtains the semi-decrypted data that only needs little decryption calculation to get the plaintext data.

Voting Service Providers: VSP is composed of the vote-counting agency and verification agency. VSP can easily decrypt the semi-decrypt data from the DSP and verify the honesty of the DSP. The verification agency can compare the decrypted data with the verification index on the blockchain to the correctness of the vote data.

B. Vote and Verifications

1) Starting Vote: the decision center collects the feedback from the management area by starting a vote. A major restriction on nodes is that only nodes with subject-related attributes can vote. After the decision center releases the decision, it starts a vote on the decision subject, including the management area and the node attributes, which also constitute the access structure.

2) Nodes Vote: after the node has executed the decision, it will evaluate and vote according to its results. Nodes can vote by themselves when they are online. Taking into the IoVs environment, most nodes cannot remain online, we adopt the proxy vote method that the offline nodes host their vote right to the nearest, trusted online nodes with the same attributes. The specific method is as follow Fig. 4.

Online nodes vote: the nodes will feedback through voting, send the encrypted data with nodes' attributes to cloud storage and the verification index to the blockchain for notarization.

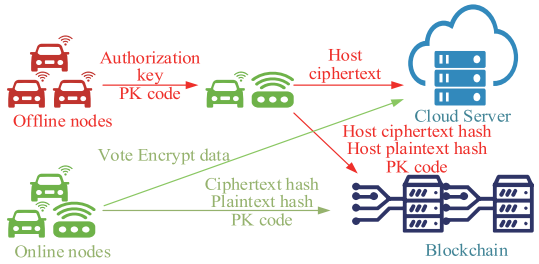


Fig. 4. Nodes vote according to their status.

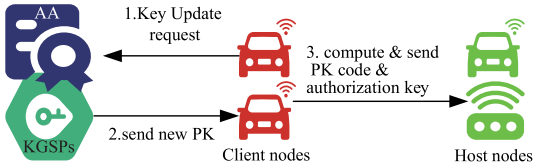


Fig. 5. The process of entrusting vote right.

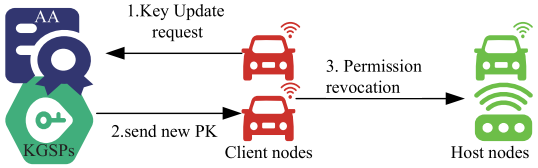


Fig. 6. The process of vote right revocation.

Proxy vote for online nodes: before the nodes go offline, their vote rights will be managed out, and the host nodes will proxy vote.

The IoVs nodes entrusting vote right is shown in Fig. 5. The nodes apply to the AA for the encryption key before going offline, and then use this key to calculate the AK. Finally, the nodes send the AK and the code of the PK to the host nodes. The host nodes are the closest nodes with the same attributes, so the view of the host nodes and the nodes are the most similar.

The revocation process is shown in Fig. 6. The nodes go back online or think the host nodes are not appropriate, they can revoke the proxy vote. The nodes first apply to AA to update the key pairs with their attribute set. After AA updates the key pairs, it sends the new keys to the nodes. The nodes recalculate the authorization keys and notify the host nodes that their authorizations have been revoked.

The vote service provider counts the vote when the IoVs nodes are voting. The VSP reads all votes, but only counts the vote related to the attribute set of the vote subject. VSP first selects the vote subject, and then determines the attribute set of the vote subject-related IoVs nodes. Then VSP applies to AA for decryption key of the attribute set and send detail of the encrypted data and outsourcing decryption key to DSP. After the DSP sends back the semi-decrypted data, VSP decrypts the semi-decryption and verify, and finally summarizes the verified voting data.

3) Verification of Outsourcing Microservices: the verification of outsourcing microservices mainly relies on the verification index on the blockchain. The vote nodes send the verification data to the blockchain to ensure the traceability and tamper-proof of the verification data. Therefore, collusion between outsourcing service providers will be easily checked out by comparing the data received with the data on the blockchain.

a) Verification of KGSPs work: AA outsources the key generation to two KGSPs, and these two providers do not know which part of the work they assign. AA can easily verify the correctness of the work of KGSPs by comparing the results of the KGSPs. Only after passing the verification can the key be generated.

b) Verification of CSS data: DSP uses the voting details got from VSP to request the attribute-related vote encryption data from the cloud storage server. Then DSP compares the encrypted data sent by CSS with the hash of the encrypted data on the blockchain to verify the correctness of the received data.

c) Verification of DSP work: The verification module in VSP decrypts the semi-decrypted data from DSP to obtain the plaintext, and then use the plaintext hash on the blockchain to verify the accuracy of the plaintext.

IV. Proxy Vote and Revocation Scheme Based on Outsourcing Microservices

Revocation of proxy vote means that the IoVs nodes apply to AA to stop sending decryption key on their request. After the revocation, the data encrypted by the previous authorization key cannot be decrypted. So, the revocation makes the host nodes become unauthorized nodes.

In the process of outsourcing key generation, the attribute set requested by the nodes will be mixed with the additional version attribute υ generated by nodes. The nodes will update their version attributes when they request for update key. The hybrid key strategy can complete the outsourcing key generation, and quickly verify the authorization key. The low version key will be excluded when it encounters the high version key.

This paper adopts the method in [23] and sets a blinding factor t to blind transformation key (TK). This blind TK can be sent to the DSP to decrypt part of the ciphertext (CT) without the participation of the secret key (SK). Therefore, the SK or plaintext data will not be leaked.

A. Preliminary

Definition 1 (Access Structure): let $\{P_1, P_2, P_3, \ldots, P_n\}$ be a set of participants. If for any sets A, B, C satisfy $B \in \mathrm{A}, B \subseteq C$, then $C \in A$ and the set $\mathrm{A} \subseteq 2^{\{P_1, P_2, P_3, \ldots, P_n\}}$ is monotonous. An access structure that is the non-empty subsets of $\{P_1, P_2, P_3, \ldots, P_n\}$ in A is defined as authentication sets. Furthermore, the predicate $\gamma(\bullet, \bullet)$ is defined as follows:

$$\gamma(\omega, \mathrm{A}) = \begin{cases} 1, & \omega \in \mathrm{A} \\ 0, & others \end{cases} \tag{1}$$

Access structure in this paper is described as $\mathrm{A} = \{\omega \subseteq U : |\omega \cap \omega^*| \geq d\}$ where ω, ω^* are attributes, U is attribute universe and d is threshold.

Definition 2 (Access Tree): in an access tree Γ, r means root node. Every leaf node m is a threshold and is denoted as (k_m, num_m). num_m means the number of m node's child node. k_m means the value of this threshold. Define three functions on the access tree Γ as follows.

- parent(m): return the parent node of this node m.
- att(m): return the attribute of this node m.
- index(m): return the number of this node m in its brother nodes.

Definition 3 (Bilinear Map): let q be a large prime, G, G_T be multiplicatively cyclic groups of q order and g is a generator of G. The map of bilinear pairings is denoted as $e : G \times G \to G_T$, which has the following three properties.

- Bilinearity: $\forall g_1, g_2 \in G, \forall a, b \in {}_R Z_q$, $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$.
- Non-degeneracy: $\exists g_1, g_2 \in G$, $e(g_1, g_2) \neq 1$.
- Computability: $\forall g_1, g_2 \in G$, there exists an efficient algorithm to compute $e(g_1, g_2)$.

B. Algorithm Definition of Our Scheme

The algorithm definition of the proposed scheme is shown as follows.

- **Initialization algorithm.**

$Setup(\lambda)$: AA runs the initialization algorithm with the security parameter λ and outputs a public key PK and a master key MK. Denote the attribute structure as Γ and attribute set as ω. m is the leaf node in Γ.

- **Key generation algorithm.**

$KenGen_{init}(\Gamma, MK)$: the initialization algorithm of outsourcing key generation takes the access structure Γ and the master key MK as input. It outputs a set of outsourcing keys $(OK_{KGSP[i]}, OK_{AA})$ and also outputs $(S[1]_{REAL}, S_{RG})$ and $(S[2]_{REAL}, S_{RG})$ to $KGSP[1]$ and $KGSP[2]$.

$KenGen_{out}(S[j]_{REAL}, S_{RG})$: $KGSP[j]$ runs this outsourcing key generation algorithm with input $S[j]_{REAL}, S_{RG}$, and it outputs partial transformation key $(TK_{KGSP[j]}, TK_{RG_j})$ to AA.

$KeyGen_{in}(\Gamma, OK_{AA})$: AA runs this internal key generation algorithm and takes the access structure Γ and the outsourcing key OK_{AA} as input. This algorithm outputs another partial transformation key TK_{AA}.

$KeyCheck(TK_{KGSP1}, TK_{RG_1}, TK_{KGSP2}, TK_{RG_2})$: this algorithm can help AA verify the correctness of the partial transformation keys from KGSPs. If all partial transformation keys from KGSPs pass the verification, a complete transformation key TK will be generated.

$TKBlind(TK)$: transformation key blinding algorithm takes complete transformation key TK as input and outputs secret key SK and blinded transformation key $\widetilde{TK}$.

- **Encryption algorithm.**

$Encrypt(M, PK, \omega)$: the encryption algorithm for voting data is calculated by the IoVs nodes, inputs voting data M, public key PK, attribute set ω and outputs ciphertext CT.

- **Authorization algorithm.**

$Authorize_{off}(PK, \omega)$: authorization algorithm for offline nodes entrusting vote right is calculated before the nodes go offline, takes public key PK and attribute set ω as input and outputs authorization key AK.

$Encrypt_{host}(M, AK)$: this encryption algorithm helps host nodes encrypt vote data. It takes authorization key AK, vote data M as input and outputs ciphertext CT.

- **Revocation of authorization algorithm.**

$Revoke(\Gamma)$: AA takes the access structure Γ from IoVs nodes and outputs the new key pair with the updated version.

- **Decryption algorithm.**

$Decrypt_{out}(CT, \widetilde{TK})$: outsourcing decryption algorithm is run by DSP. It takes ciphertext CT, blinded transformation key $\widetilde{TK}$ as input and outputs partially decrypted ciphertext CT_{part} when right $\widetilde{TK}$, otherwise outputs $\perp$.

$Decrypt_{VSP}(CT_{part}, SK)$: the decryption algorithm is run by VSP. It takes the partially decrypted ciphertext CT_{part}, secret key SK as input and outputs the original vote data when right SK, otherwise outputs $\perp$.

C. Algorithm Construction of Our Scheme

- **Initialization algorithm.**

$Setup(\lambda)$: let the first n elements in Z_q be the universe and $Z_q = (i.e. 1, 2, \ldots, n \bmod q)$. Then pick a generator $g \in {}_R G$ and an integer $x \in {}_R Z_q$, and let $g_1 = g^x$. Select elements $g_2, h, h_1, .., h_n \in {}_R G$ and finally generate the public key $PK = (g, g_1, g_2, h, h_1, .., h_n)$ and master key $MK = x$.

- **Key generation algorithm.**

$KenGen_{init}(\Gamma, MK)$: for each key generation request of node's access structure Γ, AA selects $x_{11}, x_{12} \in {}_R Z_q$ and sets $OK_{KGSP[1]} = x_{11}$, $OK_{KGSP[2]} = x_{12}$ and $OK_{AA} = x_2 = x - x_{11} - x_{12} \bmod q$. Traverse the access structure from top to bottom and select $(d_m - 1) - degree$ random polynomials $q_{KGSP[1]}(\cdot)$ and $q_{KGSP[2]}(\cdot)$ with the following restrictions.

1) $q_{KGSP[1]}(i) = q_{KGSP[2]}(i)$ where $i \in \omega$.
2) $q_{KGSP[1]}(0) = x_{11}, q_{KGSP[2]}(0) = x_{12}$ where $m = r$.

Randomly select a polynomial $g_{RG}(\cdot)$ for verifying the correctness of KGSPs' work, and choose $r_{KGSP[1].i}$, $r_{KGSP[2].i}, r_{RG.i} \in {}_R Z_p$, $r_{KGSP[1].i} = r_{KGSP[2].i}$. Finally, to prevent all KGSPs from knowing which part of transformation key they received, AA must randomly send $(S[j]_{REAL}, S_{RG})$ to $KGSP[j]_{j\in\{1,2\}}$.

$KenGen_{out}(S[j]_{REAL}, S_{RG})$: $KGSP[j]$ generates partial transformation key for $q_{KGSP[j]}(\cdot)$ and $g_{RG}(\cdot)$. $KGSP[j]$ calculates $TK_{KGSP[j]} = (\{d[j]_{i0}, d[j]_{i1}\}_{i\in\omega})$, where $d[j]_{i0} = g_2^{q_{KGSP[j]}(i)}(g_1 h_i)^{r_{KGSP[j].i}}$ and $d[j]_{i1} = g^{r_{KGSP[j].i}}$. $KGSP[j]$ calculates $TK_{RG_j} = (\{d[RG_j]_{i0}, d[RG_j]_{i1}\})$ where $d[RG_j]_{i0} = g_2^{q_{RG}(i)}(g_1 h_i)^{r_{RG.i}}$ and $d[RG_j]_{i1} = g^{r_{RG.i}}$. Finally, $KGSP[j]$ sends the results $(TK_{KGSP[j]}, TK_{RG_j})$ to AA.

$KeyGen_{in}(\Gamma, OK_{AA})$ AA randomly selects $r_v \in {}_R Z_q$ and calculates $d_{v0} = g_2^{x2}(g_1 h)^{r_v}$ and $d_{v1} = g^{r_v}$. Then AA outputs $TK_{AA} = (\{d_{v0}, d_{v1}\})$.

$KeyCheck(TK_{KGSP1}, TK_{RG_1}, TK_{KGSP2}, TK_{RG_2})$: AA checks whether all KGSPs have generated the correct output,

that is to check whether these following equations are all true.

$$\begin{cases} d[1]_{i0} = d[2]_{i0}, \\ d[1]_{i1} = d[2]_{i1}, \\ d[RG_1]_{i0} = d[RG_2]_{i0}, \\ d[RG_1]_{i1} = d[RG_2]_{i1}. \end{cases} \tag{2}$$

If these equations are all true, then AA continues to calculate $d_{i0} = d[1]_{i0} \cdot d[2]_{i0}$ and $d_{i1} = d[1]_{i1} \cdot d[2]_{i1}$. These two parts of key will be combined to output complete transformation key $TK = (\{d_{i0}, d_{i1}\}_{i\in\omega\cup\{v\}})$.

$TKBlind(TK)$: AA randomly selects $t \in {}_RZ_q$, calculates $\widetilde{TK} = (\{d_{i0}^t, d_{i1}^t\}_{i\in\omega})$ and finally outputs blinded transformation key $\widetilde{TK}$ and secret key $SK = (t, TK)$.

- **Encryption algorithm.**

$Encrypt(M, PK, \omega)$: online IoVs nodes select a random number $s \in {}_RZ_q$ and then calculate $C_0 = M \cdot e(g_1, g_2)^s$, $C_1 = g^s$, $E_v = (g_1 h)^s$, $E_i = \{(g_1 h_i)^s\}_{i\in\omega\cup\{v\}}$. The nodes finally output the ciphertext $CT = (\omega\cup\{v\}, C_0, C_1, E_v, \{E_i\}_{i\in\omega\cup\{v\}})$.

- **Authorization algorithm.**

$Authorize_{off}(PK, \omega)$: the being-offline nodes calculate the authorization key before going offline, that is, these nodes apply to AA for a set of encryption keys, then select a random number $s \in {}_RZ_q$ and calculate $A_0{=}e(g_1, g_2)^s$, $E_v = (g_1 h)^s$, $E_i = \{(g_1 h_i)^s\}_{i\in\omega}$. Finally, these nodes output the authorization key $AK = (\omega \cup \{v\}, A_0, C_1, E_v, \{E_i\}_{i\in\omega\cup\{v\}})$.

$Encrypt_{host}(M, AK)$: the host nodes help offline nodes to vote and encrypt vote data. The host nodes calculate $C_0 = M \cdot A_0$ and generate the ciphertext $CT = (\omega \cup \{v\}, C_0, C_1, E_v, \{E_i\}_{i\in\omega\cup\{v\}})$.

- **Revocation algorithm.**

$Revoke(\Gamma)$: IoVs nodes request to AA to update the key pair corresponding to the access structure Γ. AA recalculates $KeyGen_{in}(\Gamma, OK_{AA})$ with new version v and new OK_{KGSP}. So, the new $OK_{AA} = x_2' = x - x_{11}' - x_{12}'$. Then KGSPs recalculate the outsourcing key, and AA verifies and calculates the new serect key SK'. The new key pair is (PK, MK, SK') and does not change PK, MK. Eventually the ciphertext with the higher version will not be decrypted by the key with lower version.

- **Decryption algorithm.**

$Decrypt_{out}(CT, \widetilde{TK})$: assume that the blinded transformation key $\widetilde{TK}$ contains access structure Γ. Define the following two sets for the nodes m of Γ.

S_m: the set of all child nodes of m.

$S_m' = \{i | m' \in S_m, i = index(m')\}$: the number set of all child nodes of m.

Define a recursion algorithm $DecryptNode(CT, \widetilde{TK}, m)$, where input is ciphertext $CT = (\omega \cup \{v\}, C_0, C_1, E_v, \{E_i\}_{i\in\omega\cup\{v\}})$, blinded transformation key $\widetilde{TK}$ and nodes m of access structure Γ.

If m is a leaf node, DSP uses the recursion algorithm $DecryptNode(CT, \widetilde{TK}, m)$ to compute, and the output is recorded as F_m.

$$F_m = \begin{cases} \dfrac{e\left(g^s, g_2^{t\left(q_{KGSP[1]}(0)+q_{KGSP[2]}(0)\right)} E_i^{sth_i}\right)}{e\left(g^{h_i}, E_i^s\right)} \\ \qquad = e\left(g, g_2\right)^{st\times q_m(0)}, \quad i \in \omega \\ \perp, \ others \end{cases} \tag{3}$$

If m is not a leaf node, DSP uses the recursion algorithm $DecryptNode(CT, \widetilde{TK}, m')$ for all child nodes m' of m to compute F_m.

$$\begin{aligned} F_m &= \prod_{m'\in S_m} F_{m'}^{\Delta i, S_m'(0)} \\ &= \prod_{m'\in S_m} \left(e\left(g, g_2\right)^{st\times q_{m'}(0)}\right)^{\Delta i, S_m'(0)} \\ &= \prod_{m'\in S_m} \left(e\left(g, g_2\right)^{st\times q_{parent(m')}(\mathrm{index}(m'))}\right)^{\Delta i, S_m'(0)} \\ &= \prod_{m'\in S_m} \left(e\left(g, g_2\right)^{st\times q_m(i)}\right)^{\Delta i, S_m'(0)} \\ &= e\left(\mathrm{g}, \mathrm{g}_2\right)^{st\times q_m(0)} \end{aligned} \tag{4}$$

If the attribute set in ciphertext meets the access structure, the recursion algorithm calculates and returns part of ciphertext.

$$\begin{aligned} CT_{pat} &= \frac{e\left(C_1, d_{v0}^t\right)}{e\left(d_{v1}^t, E_v\right)} \times \mathrm{DecryptNode}(CT, \widetilde{TK}, r) \\ &= e\left(g, g_2\right)^{stx_1} e\left(g, g_2\right)^{stx_2} \\ &= e\left(g_1, g_2\right)^{st} \end{aligned} \tag{5}$$

$Decrypt_{VSP}(CT_{part}, SK)$: VSP uses the secret key SK to decrypt partial decryption data received from DSP and completely decrypt the ciphertext. VSP verifies the correctness of the plaintext and the honesty of the DSP through the data on the blockchain, and the vote data passes the verification can be counted.

$$\begin{aligned} \frac{C_1}{\left(C_{pat}\right)^{\frac{1}{t}}} &= \frac{M \times e\left(g_1, g_2\right)^s}{\left(e\left(g_1, g_2\right)^{st}\right)^{\frac{1}{t}}} \\ &= \frac{M \times e\left(g_1, g_2\right)^s}{e\left(g_1, g_2\right)^s} \\ &= M \end{aligned} \tag{6}$$

D. Security Model

This article only considers unauthorized adversaries. In case of only discussing the security of ciphertext, the game between adversary $\mathcal{A}$ and simulator $\mathcal{S}$ is described as follows for the scheme proposed.

- **Initialization.**

Adversary $\mathcal{A}$ determines an attribute set ω to simulator $\mathcal{S}$.

- **Setup.**

Let SP be the system parameters. The simulator $\mathcal{S}$ runs the setup algorithm and key generation algorithm and sends public key PK to adversary.

- **Phase 1.**

Adversary $\mathcal{A}$ queries decryption for a ciphertext with access structure that are not part of the attribute set ω.

The simulator $\mathcal{S}$ runs the decryption algorithm and sends the result to adversary.

- **Challenge.**

The adversary $\mathcal{A}$ generates two distinct messages m_0, m_1 from the same message space. The simulator randomly selects $\beta \in \{0,1\}$ and encrypts M_β. Then simulator sends the result M_β to adversary.

- **Phase 2.**

The simulator $\mathcal{S}$ responds to the decryption as the same way in Phase 1.

- **Guess.**

The adversary $\mathcal{A}$ generates a guess $\beta' \in \{0,1\}$ on β of M_β. If $\beta = \beta'$, the adversary $\mathcal{A}$ wins.

- **Definition** (IND-CPA security).

If there is no the adversary $\mathcal{A}$ who can break the above security model with non-negligible advantage ε in polynomial time t, then the scheme in this article is indistinguishable against chosen-plaintext attack.

Proof: Assuming that an adversary $\mathcal{A}$ can break the solution in this paper under the IND-CPA security model with a non-negligible advantage ε in probability polynomial time, then a simulator $\mathcal{S}$ can be created with the adversary $\mathcal{A}$'s algorithm to solve the DBDH assumption.

Assume that simulator $\mathcal{S}$ randomly flips a binary coin $\mu \in \{0,1\}$. The four-tuple is $(X = g^x, Y = g^y, Z = g^z, T = e(g,g)^{xyz})$ when $\mu = 0$. Otherwise, it is a random quaternion $(X = g^x, Y = g^y, Z = g^z, T = e(g,g)^v)$ where $x, y, z, v \in Z_q$. Simulator $\mathcal{S}$ outputs μ' as a guess of μ.

Initialization: the simulator $\mathcal{S}$ runs the attack algorithm of adversary $\mathcal{A}$ and accepts the challenge attribute set ω^*.

Setup: the simulator $\mathcal{S}$ distributes the public key as follows: set $g_1 = X, g_2 = Y, h = g_1^{-1} g^{-\alpha}$ where $\alpha \in_R Z_q$. For each $i \in \omega^*$, select $\alpha_i \in_R Z_q$ and set $h_i = g_1^{-1} g^{\alpha_i}$. For each $i \notin \omega^*$, select $\alpha_i \in_R Z_q$ and set $h_i = g^{\alpha_i}$. Finally, the simulator $\mathcal{S}$ sends the public key $PK = (g, g_1, g_2, h, h_1, \ldots, h_n)$ to the adversary $\mathcal{A}$.

Phase 1: adversary $\mathcal{A}$ asks the key of the access structure Γ with the restriction that ω^* does not satisfy the access structure Γ. The process of simulator $\mathcal{S}$ generating a key for adversary $\mathcal{A}$ is as follows:

Root node: $KGSP[j]_{j\in\{1,2\}}$ firstly define a $(d_m - 1) - degree$ random polynomial $q_{KGSP[j],m}(\cdot)$ for the root node m. The constant of q_m is $q_{KGSP[j],r}(0) = x_{1\ j}$, randomly select $q_{KGSP[j]}(i) \in Z_q$ as other coefficients where $i = att(m)$. Let m' be a child node of m, if $\Gamma_{m'}(\omega^*) = 1, q_{KGSP[j],r}\left(\text{index}\left(m'\right)\right)$ is known and $d[j]_{i0} = g_2^{q_{KGSP[j]}(i)} (g_1 h_i)^{r_{KGSP[j].i}}, d[j]_{i1} = g^{r_{KGSP[j].i}}$; if $q_{KGSP[j],r}\left(\text{index}\left(x'\right)\right)$ is unknown, $q_{KGSP[j],r}(0) = x_{1\ j}$ is known, $g^{q_{KGSP[j],r}(\text{index}(x'))}$ can be calculated by Lagrangian interpolation on the index. $d[j]_{i0} = g_2^{\sum_{m\in\Gamma'} \Delta_{j,S}(i) q_{KGSP[j]}(m) - (x_2 + x_{1\ j} + \alpha_i)\Delta_{m,S}(0)} (g_1 h_i)^{r'_i}, d[j]_{i1} = g^{-y\Delta_{m,S}(0) + r'_i}$ where $r_{\text{KGSP}[j],i} = -y\Delta_{m,S}(0) + r'_i$ and $r'_i \in_R Z_q$.

Internal Node: if $q_{KGSP[j],parent(m)}(\text{index}(m))$ is known, then randomly select $(k_m - 1)$ points, combine $\left(0, q_{KGSP[j],parent(m)}(\text{index}(m))\right)$ to determine the unique $q_{KGSP[j],m}(\cdot)$; if $q_{KGSP[j],parent(m)}(\text{index}\, m))$ is unknown, $g^{q_{KGSP[j]} r(\text{index}(m))}$ can be calculated by combining with the root node, then use the method in the root node to determine the unique $q_{KGSP[j],m}(\cdot)$.

The key corresponding to each leaf node m is as follows, where $i = att(m)$: when $i \in \omega^*$, $d[j]_{i0} = g_2^{q_{KGSP[j]}(0)} g_2^{q_{KGSP[j]}(i)} (g_1 h_i)^{r_{KGSP[j].i}}$, $d[j]_{i1} = g^{r_{KGSP[j].i}}$, where $d[j]_{i0}$

$$
\begin{aligned}
&= g_2^{\sum_{m\in\Gamma'} \Delta_{j,S}(i) q_{KGSP[j]}(m) - (x_2 + x_{1\ j} + \alpha_i)\Delta_{m,S}(0)} \\
&\quad \times (g_1 h_i)^{r'_i} \\
&= g_2^{\sum_{m\in\Gamma'} \Delta_{j,S}(i) q_{KGSP[j]}(m) - (x_2 + x_{1\ j} + \alpha_i)\Delta_{m,S}(0)} \\
&\quad \times (g_1 h_i)^{r_{\text{KGSP}[j],i} + y\Delta_{m,S}(0)} \\
&= g_2^{\sum_{m\in\Gamma'} \Delta_{j,S}(i) q_{KGSP[j]}(m) - (x_2 + x_{1\ j} + \alpha_i)\Delta_{m,S}(0)} \\
&\quad \times (g_1 h_i)^{r_{\text{KGSP}[j],i}} g_2^{(x+\alpha_i)} \\
&= g_2^{\sum_{m\in\Gamma'} \Delta_{j,S}(i) q_{KGSP[j]}(m) + (x - x_2 - x_{1\ j})\Delta_{m,S}(0)} \\
&\quad \times (g_1 h_i)^{r_{\text{KGSP}[j],i}} \\
&= g_2^{x - x_2 - x_{1j}} g_2^{q_{KGSP[j]}(i)} (g_1 h_i)^{r_{KGSP[2].i}}
\end{aligned} \tag{7}
$$

And now output $TK = \left(\{d_{i0}, d_{i1}\}_{i\in\omega^*}\right)$, where $d_{i0} = d[1]_{i0} \times d[2]_{i0} = g_2^{x_{12}+x_{11}} g_2^{2\times q_{KGSP[j]}(i)} (g_1 h_i)^{2\times r_{KGSP[j].i}}$, $d_{i1} = d[1]_{i1} \times d[2]_{i1} = g^{2\times(-y\Delta_{m,S}(0) + r'_i)} = g^{2\times r_{KGSP[j].i}}$.

Then the simulator $\mathcal{S}$ combines $KenGen_{\text{in}}(\Gamma, OK_{AA})$ and $KeyBlind(TK)$ algorithm to output the final private key SK. Therefore, the key generated by the simulator $\mathcal{S}$ is consistent with the key distribution generated in the original scheme.

Challenge: the adversary $\mathcal{A}$ sends two challenge messages M_0, M_1, simulator $\mathcal{S}$ throws a binary random coin to choose $\beta \in \{0,1\}$ and sets challenge ciphertext $CT^* = \left(\omega^* \cup \{v\}, M_\beta, g^z, g^{-z\alpha}, \{g^{z\alpha_i}\}_{i\in\omega^*\cup\{v\}}\right)$. If $\beta = 1$, $T = e(g,g)^{xyz}, C\ T^*$ is a valid ciphertext encrypted with a quadruple $D = (X = g^x, Y = g^y, Z = g^z, e(g,g)^{xyz})$. Otherwise, $\beta = 0, T = e(g,g)^w$, where $w \in_R Z_q$. And CT^* is encrypted with random quaternion $= (X = g^x, Y = g^y, Z = g^z, e(g,g)^w)$. Because w is random, the ciphertext CT^* is independent of β from the adversary's point of view.

Guess: the adversary outputs a guess β' of β. If $\beta' = \beta$, the simulator $\mathcal{S}$ outputs $\mu' = 0$ and the quaternion is $T = e(g,g)^{xyz}$. Otherwise the simulator $\mathcal{S}$ outputs $\mu' = 1$ and the quaternion is $T = e(g,g)^w$.

Finally, the advantage of the simulator in this game is $adv = \Pr\left[\mu' = \mu\right] - \frac{1}{2} = \Pr\left[\mu' = \mu \mid \mu = 0\right] \times \Pr[\mu = 0]$ $\Pr\left[\mu' = 1\right] \times \Pr[\mu = 1] - \frac{1}{2} = \frac{1}{2}\left(\frac{1}{2} + \varepsilon\right) + \frac{1}{2} \times \frac{1}{2} - \frac{1}{2} = \frac{\varepsilon}{2}$.

In this game, the advantage of the simulator using the algorithm of adversary $\mathcal{A}$ to break the DBDH problem is $\frac{\varepsilon}{2}$, which is not enough to break this scheme, so this scheme is IND-CPA security. □

V. Simulations and Experiments

The experimental platform is a workstation with r7 2700 CPU, 64GB RAM, 500GB SSD. The software environment is based on Windows 10 64bit and python 3.8. We simulated up to 500 IoVs nodes on this platform. They

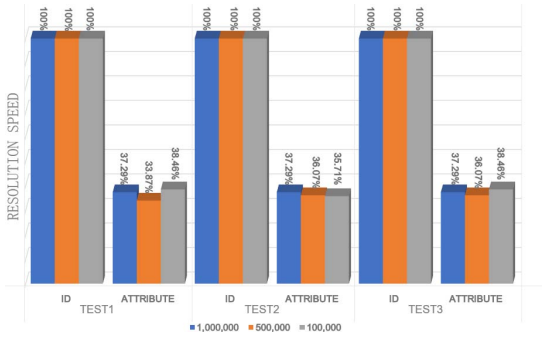


Fig. 7. The comparison of ID and attribute on resolution speed.

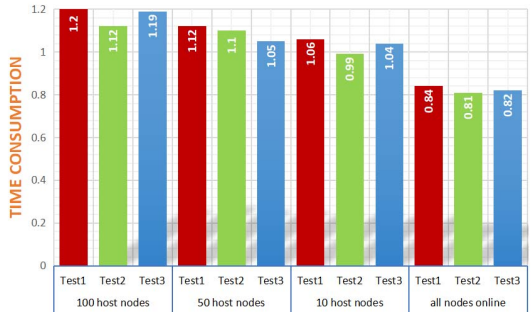


Fig. 8. The impact of the number of managed nodes on time consuming.

communicated with each other through message queuing telemetry transport (MQTT) protocol and are uniformly managed through the EMQ X enterprise. Each node subscribed to the vote topic and vote on the related topic after receiving the vote topic.

A. Comparison of Voting With ID and Voting With Attributes

In the process of vote counting, we need to verify whether the voter's information meets the target group of the vote subject, that is to verify whether the voter's attribute structures meet the attribute restrictions specified by the voting subject. Identity (ID) must query whether the information of the node meets the restriction through the ID table, but for the attributes, it can be directly verified.

We did three sets of comparison experiments on voting with ID and voting with attribute, these three experiments had 100,000 data, 500,000 data, and 1 million data respectively. The experiment shows as in Fig. 7 that the time spent checking the authority of nodes through attributes is about one-third of that through ID. Furthermore, in our scheme, the attribute set is written in ciphertext, and the voting authority can be checked without decryption, which is more efficient.

B. Simulation for Proxy Vote

In the second experiment, we simulated a smart IoVs system with voting feedback, proxy vote and 500 IoVs nodes in three situations in total. The first case is the same as the existing IoVs system, all nodes go online independently as needed and all nodes will not be online at the same time. We set the time consumption that all nodes will go online at least once as T, and all nodes are randomly online within $T = 600$ seconds. In the second case, it is an ideal situation, because it cannot stay online due to battery limitations. In this case, once the voting topic is sent out, all nodes can immediately respond and vote, and it takes very little time to complete the voting. The third case is an actual IoVs system with a proxy vote. Specifically, offline nodes entrust vote rights to trusted nodes with similar attributes before going offline, and the host node will help offline nodes vote. As in the first case, all nodes will randomly go online for a while within T time. If nodes receive the voting subject during the online state, they will vote by themselves; if nodes do not receive the voting subject during the online state, the voting rights will be managed.

We conducted three experiments, and the specific experimental results are shown in Fig. 8. It takes T time for all nodes to vote in the existing IoVs system, which is the same time consumption for all nodes to go online once. In the second ideal case, all nodes are online, and the time required for all nodes to complete the vote is extremely short. Our scheme adds the proxy vote to the existing IoVs system, and the experimental result shows our improvements make the time consumption for the existing IoVs system to complete the vote similar to that for the ideal case.

Moreover, we also evaluated the effect of the number of the online nodes in the system on the time spent by the IoVs system with proxy vote to complete voting. In this experiment, we adjusted the randomness of online nodes online in the time T so that they can keep a certain number of nodes online at the same time and become trusted nodes. We did three sets of experiments, and each set of experiments was done three times. The experimental results show that the more host nodes online at the same time, the longer it takes for all nodes to complete voting.

To sum up, our experiments did not pay attention to the efficiency of outsourcing services, because powerful equipment can greatly reduce computing time. We think that the security of vote data is more important, so our scheme uses a verification index on the blockchain as the basis for outsourcing data verification, thereby realizing the immutability of data. We use vote as feedback, and the most important thing for feedback is speed and accuracy. The proxy vote method can make the vote efficiency of the existing IoVs system close to the ideal system with all nodes online. But proxy vote means that the offline nodes cannot express their opinions, but instead use the opinions of the host nodes, which will affect the accuracy of feedback.

VI. Conclusion

To deal with the problem that the existing decision feedback of smart IoVs is limited to deep learning data and separated from the real IoVs in space and time. This paper proposes a blockchain-based vote scheme for decision-making feedback of smart IoVs to simplify and generalize the vote process by using an ABE scheme to achieve anonymous voting feedback of nodes. To overcome the resource constraints of IoVs nodes, this paper uses microservices to outsource key generation to KGSPs, decryption to DSP and storage to cloud server. Microservices will improve the concurrent performance and stability of the entire system. Vote rights can be proxied when nodes go offline. Moreover, the VSP completes the

verification of vote data and vote counting. The blockchain is used to notarize vote data and ensure the authenticity and integrity of the vote data. Finally, the experiment results show that the spending time of the IoVs system with proxy vote completing vote is very close to the IoVs system with all nodes online. Besides, the more online nodes for proxy nodes, the less spending time of voting for the whole system, but this also brings the accuracy of feedback, which will affect the precision of feedback.

REFERENCES

[1] H. Chai, S. Leng, Y. Chen, and K. Zhang, "A hierarchical blockchain-enabled federated learning algorithm for knowledge sharing in internet of vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 3975–3986, Jul. 2021.

[2] M. Sepulcre and J. Gozalvez, "Heterogeneous V2V communications in multi-link and multi-RAT vehicular networks," *IEEE Trans. Mobile Comput.*, vol. 20, no. 1, pp. 162–173, Jan. 2021.

[3] X. Wen, J. Chen, Z. Hu, and Z. Lu, "A p-opportunistic channel access scheme for interference mitigation between V2V and V2I communications," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 3706–3718, May 2020.

[4] K. Xiong, S. Leng, J. Hu, X. Chen, and K. Yang, "Smart network slicing for vehicular fog-RANs," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3075–3085, Apr. 2019.

[5] J. Wang, Y. Yang, T. Wang, R. S. Sherratt, and J. Zhang, "Big data service architecture: A survey," *J. Internet Technol.*, vol. 21, no. 2, pp. 393–405, 2020.

[6] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Blockchain empowered asynchronous federated learning for secure data sharing in internet of vehicles," *IEEE Trans. Veh. Technol.*, vol. 69, no. 4, pp. 4298–4311, Apr. 2020.

[7] Y. Ren *et al.*, "Data query mechanism based on hash computing power of blockchain in Internet of Things," *Sensors*, vol. 20, no. 1, p. 207, Dec. 2019.

[8] L. Fang *et al.*, "THP: A novel authentication scheme to prevent multiple attacks in SDN-based IoT network," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 5745–5759, Jul. 2020.

[9] S. R. Pokhrel and J. Choi, "Federated learning with blockchain for autonomous vehicles: Analysis and design challenges," *IEEE Trans. Commun.*, vol. 68, no. 8, pp. 4734–4746, Aug. 2020.

[10] X. Zhang, K.-K. R. Choo, and N. L. Beebe, "How do I share my IoT forensic experience with the broader community? An automated knowledge sharing IoT forensic platform," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6850–6861, Apr. 2019.

[11] A. R. Heravi and G. A. Hodtani, "A new correntropy-based conjugate gradient backpropagation algorithm for improving training in neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 12, pp. 6252–6263, Dec. 2018.

[12] S. Salehkaleybar, A. Sharif-Nassab, and S. J. Golestani, "Distributed voting/ranking with optimal number of states per node," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 1, no. 4, pp. 259–267, Dec. 2015.

[13] A. Paul, D. P. Mukherjee, P. Das, A. Gangopadhyay, A. R. Chintha, and S. Kundu, "Improved random forest for classification," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 4012–4024, Aug. 2018.

[14] N. Samet, S. Hicsonmez, and E. Akbas, "HoughNet: Integrating near and long-range evidence for bottom-up object detection," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Cham, Switzerland, 2020, pp. 406–423.

[15] M. R. Clarkson, S. Chong, and A. C. Myers, "Civitas: Toward a secure voting system," in *Proc. IEEE Symp. Secur. Privacy (SP)*, Oakland, CA, USA, May 2008, pp. 354–368.

[16] S. Lin, R. Zhang, H. Ma, and M. Wang, "Revisiting attribute-based encryption with verifiable outsourced decryption," *IEEE Trans. Inf. Forensics Secur.*, vol. 10, no. 10, pp. 2119–2130, 2017.

[17] J. Li *et al.*, "An efficient attribute-based encryption scheme with policy update and file update in cloud computing," *IEEE Trans. Ind. Informat.*, vol. 15, no. 12, pp. 6500–6509, Dec. 2019.

[18] J. Li, X. Huang, J. Li, X. Chen, and Y. Xiang, "Securely outsourcing attribute-based encryption with checkability," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 8, pp. 2201–2210, Aug. 2014.

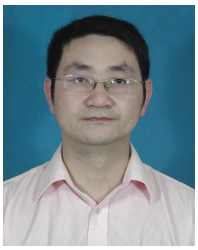
[19] W.-Y. Chiu and W. Meng, "EdgeTC—A PBFT blockchain-based ETC scheme for smart cities," *Peer-Peer Netw. Appl.*, vol. 2021, no. 3, pp. 1–13, Mar. 2021.

[20] O. Novo, "Blockchain meets IoT: An architecture for scalable access management in IoT," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1184–1195, Apr. 2018.

[21] Y. Ren *et al.*, "Multiple cloud storage mechanism based on blockchain in smart homes," *Future Gener. Comput. Syst.*, vol. 115, pp. 304–313, Feb. 2021.

[22] M. Hasan, S. Mohan, T. Shimizu, and H. Lu, "Securing vehicle-to-everything (V2X) communication platforms," *IEEE Trans. Intell. Vehicles*, vol. 5, no. 4, pp. 693–713, Dec. 2020.

[23] P. Golle and I. Mironov, "Uncheatable distributed computations," in *Proc. Cryptograph. Track RSA Conf.*, San Francisco, CA, USA, Apr. 2001, pp. 425–440.



Yongjun Ren (Member, IEEE) received the M.S. degree from Hohai University, China, in 2004, and the Ph.D. degree from the Computer and Science Department, Nanjing University of Aeronautics and Astronautics, China, in 2008. He is currently serving as a full time faculty with the Nanjing University of Information Science and Technology. His research interests include network security and applied cryptography. He is a member of ACM.

Fujian Zhu is currently pursuing the master's degree with the School of Computer and Software, Nanjing University of Information Science and Technology, with focus on applied cryptography, the Internet of Things, and blockchain. He has published some blockchain technical research articles.

Jin Wang (Senior Member, IEEE) received the B.S. and M.S. degrees from the Nanjing University of Posts and Telecommunications, China, in 2002 and 2005, respectively, and the Ph.D. degree from Kyung Hee University Korea, in 2010. He is currently a Professor with the Department of Computing Science, University of Aberdeen, Aberdeen, U.K. His research interests include wireless communications and networking, performance evaluation, and optimization. He is a member of the ACM.

Pradip Kumar Sharma (Senior Member, IEEE) received the Ph.D. degree in computer science and engineering (CSE) from the Seoul National University of Science and Technology, South Korea, in August 2019. He is currently an Assistant Professor in cybersecurity with the Department of Computing Science, University of Aberdeen, U.K. He has published many technical research articles in leading journals of the IEEE, Elsevier, Springer, and MDPI. His current research interests include cybersecurity, blockchain, edge computing, SDN, SNS, and the IoT security. He is currently an Associate Editor of *Human-Centric Computing and Information Sciences* (HCIS) and the *Journal of Information Processing Systems* (JIPS).

Uttam Ghosh (Senior Member, IEEE) received the Ph.D. degree in electronics and electrical engineering from IIT Kharagpur, India, in 2013. He is currently working as an Assistant Professor of practice with the Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN, USA. He has published 50 articles in reputed international journals, including the IEEE TRANSACTIONS, Elsevier, Springer, IET, Wiley, InderScience, and IETE, and in top international conferences sponsored by the IEEE, ACM, and Springer. His main research interests include cybersecurity, computer networks, wireless networks, information centric networking, and software defined networking. He is a member of AAAS, ASEE, ACM, and Sigma Xi.