Secure Traffic Data Sharing in UAV-Assisted VANETs

Yilin Liu[1], Yujue Wang[2], Chen Yi[4(✉)], Yong Ding[3,4], Changsong Yang[3], and Huiyong Wang[1]

[1] School of Mathematics and Computing Science, Guilin University of Electronic Technology, Guilin 541004, China
[2] Hangzhou Innovation Institute of Beihang University, Hangzhou 310000, China
[3] Laboratory of Cryptography and Information Security, School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin 541004, China
[4] Institute of Cyberspace Technology, HKCT Institute for Higher Education, Hong Kong 999077, China
alexyi@ctihe.edu.hk

Abstract. Aiming at the issues of low comprehensiveness and timeliness of data, difficulty in balancing data anonymity and traceability, and challenges of securely storing massive data in traditional traffic data sharing systems, this paper proposes a UAV-VANET integrated system (UVIS) based on consortium blockchain. The UAV integrated into the VANET can promptly provide drivers and traffic managers with comprehensive traffic information and images for traffic planning, thus enhancing transportation efficiency and safety. To achieve traceability of anonymous data sharing, we introduce a proxy re-encryption mechanism to realize precise data access control, which can not only protect data and identity privacy but also trace the true identity of malicious users. Additionally, it effectively prevents the collusion between proxies and data requesters from stealing unauthorized confidential information. To alleviate the pressure of traffic data storage, we adopt a storage method that combines blockchain and IPFS, ensuring secure storage of massive data. Security analysis shows that the UVIS has achieved secure sharing of traffic data. We analyze its efficiency theoretically, and demonstrate the practicality through experiments.

Keywords: UAV-VANET integrated system · data sharing · blockchain · privacy preservation · proxy re-encryption

1 Introduction

With the acceleration of urbanization, traditional traffic data sharing systems are confronted with many challenges. Limited access to timely, comprehensive, and multi-perspective traffic data hinders drivers from fully comprehending the increasingly complex traffic conditions, thus impeding effective traffic management and planning. At the same time, during traffic data sharing, it is difficult

H. Gao et al. (Eds.): CollaborateCom 2023, LNICST 562, pp. 301–320, 2024.
https://doi.org/10.1007/978-3-031-54528-3_17

to trace real identities under the premise of protecting data and identity privacy, making it impossible to effectively supervise and penalize malicious users, thus affecting the data quality and development of data sharing. In addition, the secure storage of massive traffic data requires substantial storage resources, so a suitable secure storage method is needed to ensure the integrity and reliability of data. This makes us continuously explore traffic data sharing systems that are more suitable for modern smart cities. In the future development of smart traffic, the collection, privacy protection, secure sharing, and secure storage of traffic data are important components of an intelligent traffic data sharing system.

The Vehicular Ad Hoc Network (VANET) has emerged as a crucial component of smart traffic and intelligent travel, making it one of the fastest developing technologies in Intelligent Transportation Systems (ITS) [2]. Its wide applications in intelligent navigation, vehicle positioning, and traffic information collection have significantly contributed to transportation intelligence, reducing traffic accidents and improving road safety [20,23]. The Unmanned Aerial Vehicle (UAV), characterized by its small size, easy deployment, and high flexibility [3], can conveniently capture real-time traffic data and images from an aerial perspective, conduct road patrols, and monitor traffic [13]. Their applications in traffic management are gradually maturing. Integrating the UAV into VANET will conduce to a more accurate and comprehensive understanding of vehicle and traffic conditions. This enables more efficient traffic flow management, alleviation of urban traffic congestion, reduction of traffic accidents, and enhancement of transportation safety. The UAV-VANET integrated system can support decision-making in vehicle navigation, traffic flow control, and road safety, thus facilitating various application scenarios of UVIS such as traffic monitoring, intelligent navigation, environmental monitoring, and emergency response.

In the UVIS that we construct, the data collected by the UAV comes from two primary sources. First, vehicle-related data, which often has a certain degree of privacy, should not involve other vehicles. Second, external environmental data, such as terrain, traffic flow, and traffic condition, is generally public information. The UAV implements access control on different types of traffic data it collects. First, the on-board unit (OBU) of the vehicle applies to the UAV for its own vehicle-related data and external environmental data to control the real-time traffic situation in an all-round way, reducing the burden of the vehicle networking communication base station GS. Second, the GS applies to the UAV for external environmental data and carries out high-level data processing and analysis of road traffic data from a macro perspective. Subsequently, the GS sends traffic instructions to communication devices such as the OBU and RSU, to achieve control of the whole road and traffic management. The two data-sharing patterns are designed from the micro and macro perspectives respectively, and the synergistic operation can achieve better traffic control effects.

1.1 Related Techniques

In the process of traffic data sharing, message accuracy, user privacy, and access control are critical factors that impact VANET service provision [9]. Due to the

sensitivity of certain data, such as vehicle location, vehicle sensor data, vehicle images, and vehicle videos, unauthorized access may lead to data leakage, data tampering, or data loss. These security breaches will have detrimental effects on VANET decision-making and operations, thus endangering the safety of vehicles and traffic. Failure to guarantee data accuracy and user privacy not only jeopardizes lives and property but diminishes user engagement [22]. However, a reliable data sharing scheme for VANET is always a great challenge.

In recent years, several solutions have been proposed to address the issue of secure data sharing in VANET. Li et al. [11] proposed the FADB scheme, which combines blockchain and CP-ABE algorithms to enable fine-grained access control and distributed storage in VANET. However, schemes based on attribute encryption always have high computational costs. Liu et al. [15] introduced a security-aware information propagation model based on attribute encryption and proxy re-encryption (PRE) for access control in VANET. Han et al. [7] protected vehicle identity privacy through a broadcast proxy re-encryption scheme with cubic spline interpolation. Wang et al. [21] designed a public key re-encryption scheme based on ciphertext delegation equality testing (PRE-DET), allowing users to share outsourced data. Eltayieb et al. [4] proposed a certificateless proxy re-encryption access mechanism for data outsourcing computation. Noh et al. [16] presented a secure data sharing system based on blockchain. But their proxy re-encryption scheme poses the risk of collusion between the cloud server and data requester to obtain the data owner's private key. The above proxy re-encryption-based data sharing schemes provide inspiration for data sharing in VANET, but do not maintain the balance between user identity privacy and traceability well and prevent collusion attacks. A secure VANET system should ensure user identity privacy protection and traceability [24].

Blockchain is an innovative technology that utilizes a chained data structure composed of chronologically connected data blocks. It employs distributed ledger and cryptographic techniques to ensure data immutability and prevent forgery [12]. Through distributed networks and consensus mechanisms, it achieves decentralized data verification, ensuring that all participating nodes can verify data authenticity [14]. The typical features of blockchain technology include decentralization, collective maintenance, resistance to single-point attacks, and immutability, which can effectively solve the problems of centralization, mutual distrust among entities, and privacy leakage in traditional VANET [1]. Therefore, blockchain establishes a trustworthy foundation for participants in untrusted environments, promoting safer, more reliable, and efficient collaborations [14].

However, the data generated by vehicles is becoming increasingly fine-grained and complex [8], leading to a sharp increase in the amount of traffic data, which causes a storage bottleneck for blockchain. Whether storage resources can meet the actual demand will be a challenge [5]. To address the issue of limited storage capacity, it is feasible to combine blockchain with IPFS (InterPlanetary File System). This combination allows for secure storage of traffic data through a collaborative on-chain and off-chain storage model. The IPFS provides a peer-to-peer (P2P) distributed storage structure [10], which can effectively avoid single-

point failures of centralized cloud servers [19]. Moreover, it can easily store large amounts of data, overcoming limitations of traditional traffic data storage systems, such as high deployment costs and insufficient storage resources. Simply storing the IPFS hash of the data rather than the complete traffic data in the block, the pressure on blockchain data storage can be relieved [26].

1.2 Our Contributions

To address the aforementioned shortcomings of traditional data sharing systems, we design a UAV-VANET integrated system and propose a consortium blockchain-based anonymous conditionally traceable data secure sharing scheme suitable for this system. The proposed scheme provides more reliable data support for the decision-making and operation of the vehicular network system, realizing the secure sharing and circulation protection of traffic data. The main contributions of this paper can be summarized as follows:

1. According to the characteristics of different types of data collected by UAV, UVIS utilizes an integrated approach of macro management and micro control strategies to cooperatively operate the different data sharing patterns, achieving the full use of data and more effective traffic control.
2. The UVIS adopts an anonymized identity and traceable authentication protocol, which not only protects user identity with pseudonyms but also reveals the identity of malicious nodes under certain conditions. It improves the efficiency of authentication by using batch authentication. Moreover, based on this protocol, this scheme allows for the easy implementation of key recovery.
3. The UVIS constructs a decentralized architecture based on the transportation consortium blockchain (TCB), creating a trusted environment for traffic data sharing and collaboration. To enhance storage capacity, UVIS combines blockchain technology with the IPFS distributed file system. Only the IPFS hash of the data is stored on the blockchain, while the complete data is transferred to IPFS, effectively alleviating the storage pressure on the blockchain.
4. The UVIS utilizes Proxy Re-Encryption for access control of data, which effectively protects data privacy and security, preventing data leakage and abuse. We use the random number generated by the Verifiable Random Function (VRF) to elect a leader (proxy node) and design for encryption and data access applications. This can prevent the proxy and data requester from conspiring to steal confidential information without authorization.

2 System Architecture and Security Requirements

2.1 UVIS Architecture

As shown in Fig. 1, the UAV-VANET integrated system (UVIS) consists of three main components: UAV, TCB, and IPFS.

1. UAV: The UAV transmits the collected traffic data to the vehicle networking communication base station (GS) and On-Board Unit (OBU) through wireless communication technology. It provides real-time data to the VANET, which can be used for traffic condition monitoring, road condition prediction, and traffic management.
2. TCB: The transportation consortium blockchain network comprises three types of nodes:

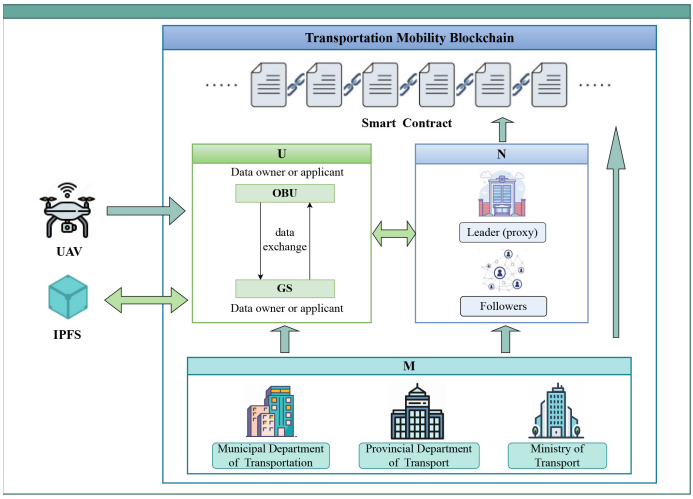

Fig. 1. System architecture of UVIS

(1) User nodes $\mathcal{U}$: $\mathcal{U}$ are composed of data owners and data requesters. In UVIS, OBU and GS are both data owners and data requesters, interacting with each other for data access. On the one hand, the OBU and GS are respectively responsible for receiving vehicle-related information and external environmental data shared by UAVs. On the other hand, the GS needs to request certain vehicle-related information from the OBU for better traffic control, and the OBU also needs to apply to the GS for specific traffic indication data after its analysis and processing to assist drivers and facilitate road traffic management.

(2) Consensus nodes $\mathcal{N}$: $\mathcal{N}$ represent the nodes participating in the consensus process. They are responsible for generating and verifying data and blocks, as well as tracing the identities of $\mathcal{U}$. They are usually authoritative institutions such as regional traffic police teams and traffic management bureaus. In the consensus mechanism, the nodes are categorized into two roles: leader and follower, and the leader performs the proxy re-encryption process acting as a proxy.

(3) Management node $\mathcal{M}$: $\mathcal{M}$ is typically controlled by the municipal transportation bureau, provincial transportation department, or national transportation department, whose duty is to manage the identity information of users $\mathcal{U}$ and play a supervisory role.

3. IPFS: IPFS adopts a decentralized storage approach to distribute data across multiple nodes in the network, which can avoid single points of failure, enhancing data storage reliability and stability. In UVIS, vehicles and GS can securely store the collected data on IPFS network nodes, ensuring data safety, reliability, and long-term preservation. In addition, IPFS employs distributed hash tables for distributed data addressing, which improves the efficiency and accuracy of data acquisition.

2.2 Security Requirements

To achieve secure and efficient data interaction in UVIS, the traffic data sharing scheme of UVIS needs to meet the following requirements:

1. Identity Privacy Protection: Due to the possibility that traffic data may contain personal privacy data such as vehicle location, many driviers are concerned about disclosing their identities.
2. Conditional Identity Tracking: When the system detects any security event, such as unauthorized access or attempts to invade the system, it needs to track the real identity of malicious user to prevent further security threats.
3. Resistance to Collusion Attacks: The leader in the consensus group, acting as a proxy node, is a curious semi-trusted entity that may collude with the data applicant to steal unauthorized traffic data.
4. User Key Recovery: In traffic data sharing, users, such as OBU and GS, may experience private keys loss owing to various factors, such as equipment failure, virus attacks, human errors, and natural disasters. The loss of private keys will disrupt the secure transmission of traffic information, thus impacting traffic efficiency and safety.
5. Secure Storage of Traffic Data: The storage capacity of the blockchain system is limited. With the increase of mass traffic data, there will be problems of insufficient storage capacity, which may bring about incomplete data storage or inability to store, affecting the security of traffic data storage.

3 The Proposed Scheme

3.1 System Initialization

Given a security parameter κ, the management node $\mathcal{M}$ generates a bilinear map $e : G_1 \times G_1 \rightarrow G_T$, where G_1 and G_T are cyclic groups of prime order q, and g_1, g_2 are two distinct generators of G_1. Then, $\mathcal{M}$ picks eight collision-resistant hash functions: $H_1 : G_1 \rightarrow \{0,1\}^*, H_2 : \{0,1\}^* \rightarrow Z_q^*, H_3 : G_1 \rightarrow Z_q^*, H_4 : G_1 \rightarrow G_1, H_5 : Z_q^* \rightarrow G_1, H_6 : G_1 \times G_1 \rightarrow Z_q^*, H_7 : G_T \times \{0,1\}^* \times Z_q^* \rightarrow$

$Z_q^*, H_8 : G_1 \times G_1 \times G_1 \times G_1 \times G_1 \times G_1 \rightarrow Z_q^*$, randomly selects the private key $sk_{mn} \in Z_q^*$, and computes the corresponding public key $pk_{mn} = {g_1}^{sk_{mn}}$. The consensus node $\mathcal{N}_c$ randomly selects a private key $sk_c \in Z_q^*$, and computes the public key $pk_c = {g_1}^{sk_c}$, $1 \leqslant c \leqslant n$. Let the leader in the consensus group be $\mathcal{N}_{\mathscr{L}}$, whose private key is $sk_{\mathscr{L}}$ and public key is $pk_{\mathscr{L}}$. The users $\mathcal{U}$ choose a secure signature scheme $\mathcal{F} = (SigGen, SigVerif)$ [17]. Finally, $\mathcal{M}$ publishes the public system parameters $params = \{\kappa, G_1, G_2, g_1, g_2, q, e, H_1, \ldots, H_8, pk_{mn}, n\}$.

3.2 User Registration

Pseudo-Identity Generation. First, $\mathcal{M}$ constructs real identity information $Info_i$ for a user $\mathcal{U}_i$. Then, $\mathcal{M}$ randomly selects $s_{1,i} \in Z_q^*$, calculates $S_{1,i} = g_1^{s_{1,i}}$, and generates identity-protected information $\pi_i = H_1(S_{1,i}) \oplus Info_i$ for $\mathcal{U}_i$.

Next, $\mathcal{M}$ selects two random numbers $\alpha_i, \beta_i \in Z_q^*$, and calculates the pseudo-identity PID_i and signature σ_i of $\mathcal{U}_i$ through the following equations:

$$z_i = \alpha_i(\beta_i + H_2(\pi_i)) \bmod q, PID_i = {g_1}^{z_i}$$
$$\delta_i = H_3(PID_i), \sigma_i = (z_i + \delta_i sk_{mn}) \bmod q$$

Finally, $\mathcal{M}$ sends the identity information $\{PID_i, \sigma_i\}$ to the user $\mathcal{U}_i$.

Key Generation. After receiving the identity information $\{PID_i, \sigma_i\}$ from $\mathcal{M}$, $\mathcal{U}_i$ computes $\delta^* = H_3(PID_i)$, and verifies:

$$g_1^{\sigma_i} = (pk_{mn})^{\delta^*} \times PID_i \tag{1}$$

If the authentication condition is satisfied, $\mathcal{U}_i$ takes PID_i as his pseudo-identity.

The user $\mathcal{U}_i$ chooses a random number $a_i \in Z_q^*$, computes $s_{2,i} = s_{1,i} - a_i, S_{2,i} = g_1^{s_{2,i}}$, and generates his own private key $sk_i = H_3(S_{2,i})$ and public key $pk_i = H_4(PID_i)^{sk_i}$. Then, $\mathcal{U}_i$ computes $A_i = {g_1}^{a_i}$, generates the signature $sig_{A,i}$ though the $F.Sign$ signature algorithm, and saves $\{a_i, A_i, sig_{A,i}\}$.

Then, $\mathcal{U}_i$ acquires the public random number $u \in Z_q^*$ locally generated by the current leader using the VRF function, picks a random number $k_i \in Z_q^*$, and computes:

$$K_i = {g_1}^{k_i}, u_i = H_5(u)^{sk_i}$$
$$\delta_{u,i} = H_2(PID_i \parallel K_i \parallel u_i)$$
$$\sigma_{u,i} = (u + \delta_{u,i} sk_i) \bmod q$$

Finally, $\mathcal{U}_i$ generates his own basic information $\{PID_i, \sigma_i, K_i, u_i, \sigma_{u,i}\}$.

3.3 Encryption

The user $\mathcal{U}_i$ chooses a random number $\omega \in \{0,1\}^*$ and encrypts traffic data m as follows:

$$D = e(H_4(PID_i), H_5(u)^{a_i})$$
$$C_1 = (m \parallel \omega) \oplus D^{H_6(PID_i \| pk_i)}$$
$$C_2 = H_4(PID_i)^u$$
$$C_3 = (u + sk_i H_7(D \parallel C_1 \parallel C_2)) \bmod q$$

Finally, $\mathcal{U}_i$ sends the ciphertext $\{D, C_1, C_2, C_3\}$ to $IPFS$.

3.4 Data Storage

When receiving $\{D, C_1, C_2, C_3\}$ from $\mathcal{U}_i$, $IPFS$ verifies the following equation:

$$H_4(PID_i)^{C_3} = C_2 \times pk_i^{H_7(D\|C_1\|C_2)} \tag{2}$$

If the verification condition is satisfied, $IPFS$ stores $\{D, C_1, C_2, C_3\}$ and generates the corresponding download address url of the ciphertext for $\mathcal{U}_i$.

3.5 Data Upchain

To alleviate the storage burden of blockchain, $\mathcal{U}_i$ initiates an on-chain request to record his metadata, which only includes the download address url, the hash value h_m of the traffic data m, and the pseudo-identity PID_i of $\mathcal{U}_i$.

After the successful block upload, the consensus group determines the leader of the next round is $\mathcal{N}_{\mathscr{L}}$ through the verifiable random number num generated by the VRF function and the formula $\mathscr{L} = (num \mod n) + 1$. The detailed process is as follows:

The current leader takes his private key sk and current timestamp x as inputs to generate a random number num and proof p. The leader publicly broadcasts the parameters $\{num, p\}$, and other follower nodes can verify whether the random number num generated by the leader is effective through the leader's public key $pk_{\mathscr{L}}$, current timestamp x, and proof p.

After all nodes pass the verification, the consensus group calculates $\mathscr{L} = (num \mod n)+1$ and elects $\mathcal{N}_{\mathscr{L}}$ as the leader. Applying the randomized output value of the VRF function to determine the leader can ensure the fairness and unpredictability of the election.

3.6 Application, Authorization and Access

If a data applicant $\mathcal{U}_j$ wants to access m of $\mathcal{U}_i$, $\mathcal{U}_j$ needs to be authorized by $\mathcal{U}_i$.

Application and Authorization. The data applicant $\mathcal{U}_j$ requests data access from $\mathcal{U}_i$ and sends his basic information $\{PID_j, \sigma_j, K_j, u_j, \sigma_{u,j}\}$. Then, $\mathcal{U}_i$ verifies the identity of $\mathcal{U}_j$ by Eq. (1), and verifies the parameter u_j:

$$H_4(PID_j)^{\sigma_{u,j}} = (pk_j)^{H_2(PID_j\|K_j\|u_j)} \times H_4(PID_j)^u$$

If the above equation holds, the verification is successful. Then $\mathcal{U}_i$ grants permission to $\mathcal{U}_j$ to access the requested data.

Re-encryption Key Generation. The user $\mathcal{U}_i$ computes:

$$d_j = PID_j \times (pk_{mn})^{H_3(PID_j)}$$
$$rk_1 = H_8({d_j}^{k_i} \parallel {K_j}^{\sigma_i} \parallel PID_i \parallel PID_j \parallel pk_i \parallel pk_j)$$
$$rk_2 = (u_j)^{\frac{a_j}{sk_i}}$$

Then, $\mathcal{U}_i$ generates the re-encryption key $r_{i\to j} = \{rk_1, rk_2\}$, uses $\mathcal{F}.Sign$ to generate signature sig_r on $r_{i\to j}$, and sends $\{r_{i\to j}, sig_r\}$ to the proxy node.

3.7 Proxy Re-encryption

Because the leader is the proxy node, the proxy node's private key is $sk_{\mathscr{L}}$ and public key is $pk_{\mathscr{L}}$. The proxy node receives $\{r_{i\to j}, sig_r\}$ and executes the $\mathcal{F}.SignVerif$ algorithm to verify the signature. Upon successful verification, it obtains the ciphertext $\{D, C_1, C_2, C_3\}$ from $IPFS$ and verifies it by equation (2). If the verification is successful, the proxy node re-encrypts the ciphertext by the following equations:

$$D' = e(pk_i, rk_2)$$
$${C_1}' = {C_1}^{rk_1}$$
$${C_2}' = {g_1}^{u}$$
$${C_3}' = (u + sk_{\mathscr{L}} H_7(D' \parallel {C_1}' \parallel {C_2}')) \bmod q$$

then generates the re-encrypted ciphertext $\{D', {C_1}', {C_2}', {C_3}'\}$, which is transmitted to $\mathcal{U}_j$.

3.8 Ciphertext Decryption

Self-decryption. The user $\mathcal{U}_i$ derives the ciphertext $\{D, C_1, C_2, C_3\}$ from $IPFS$, and verifies it with equation (2). After successful verification, the ciphertext is decrypted by the following equations:

$$D = e(H_4(PID_i), H_5(u)^{a_i})$$
$$m \parallel \omega = C_1 \oplus D^{H_6(PID_i \| pk_i)}$$

Re-decryption. After receiving the re-encrypted ciphertext $\{D', {C_1}', {C_2}', {C_3}'\}$, $\mathcal{U}_j$ then verifies it through the following equation:

$${g_1}^{{C_3}'} = {C_2}' \times {pk_{\mathscr{L}}}^{H_7(D' \| {C_1}' \| {C_2}')}$$

After successful verification, $\mathcal{U}_j$ decrypts the re-encrypted ciphertext using the following formulas:

$$d_i = PID_i \times (pk_{mn})^{H_3(PID_i)}$$
$$rk_1 = H_8({K_i}^{\sigma_j} \parallel {d_i}^{k_j} \parallel PID_i \parallel PID_j \parallel pk_i \parallel pk_j) \tag{3}$$

$$C_1 = (C_1')^{\frac{1}{rk_1}} = (m \parallel \omega) \oplus D^{H_6(PID_i \| pk_i)}$$

$$D = (D')^{\frac{1}{sk_j}}$$

$$m \parallel \omega = C_1 \oplus D^{H_6(PID_i \| pk_i)}$$

3.9 Traceability of Pseudo-identities

In this paper, a secret sharing method is employed to achieve the traceability of pseudo-identities. To share $s_{1,i}$, $\mathcal{M}$ selects $t-1$ random numbers $a_{1,i}, \ldots, a_{t-1,i} \in Z_q^*$, constructs a $(t-1)$-degree polynomial:

$$f_i(x) = a_{0,i} + a_{1,i}x + a_{2,i}x^2 + \cdots + a_{t-1,i}x^{t-1} \bmod q$$

where $a_{0,i} = s_{1,i}$, computes the polynomial shares $\{f_i(1), f_i(2), \cdots, f_i(n)\}$, and generates the commitments of polynomial shares $\langle Y_c \rangle_{c=1\sim n}$:

$$Y_c = {pk_c}^{f_i(c)}, 1 \leqslant c \leqslant n$$

For subsequent verifications of the polynomial shares, $\mathcal{M}$ also needs to compute the commitments of the polynomial parameters $\langle C_l \rangle_{l=0\sim t-1}$:

$$C_l = {g_2}^{a_{l,i}}, 0 \leqslant l < t-1$$

commitments of all polynomial shares $\langle X_c \rangle_{c=1\sim n}$:

$$X_c = {g_2}^{f_i(c)}, 1 \leqslant c \leqslant n$$

Then, $\mathcal{M}$ broadcasts $\{\pi_i, \langle C_l \rangle_{l=0\sim t-1}, \langle X_c \rangle_{c=1\sim n}, \langle Y_c \rangle_{c=1\sim n}\}$ to the entire TCB network.

With the public information $\{\pi_i, \langle C_l \rangle_{l=0\sim t-1}, \langle X_c \rangle_{c=1\sim n}, \langle Y_c \rangle_{c=1\sim n}\}$, each $\mathcal{N}_c$ can not only verify the correctness of the received polynomial commitment Y_c but also check the consistency of all polynomial commitments $\langle Y_c \rangle_{c=1\sim n}$, ensuring that $\mathcal{M}$ is honest during the distribution process of $\langle Y_c \rangle_{c=1\sim n}$. The specific verification steps are as follows:

First, to check whether $f_i(c)$ in the commitment X_c is generated by the polynomial $f_i(x)$ constructed by $\mathcal{M}$, $\mathcal{N}_c$ verifies the following equation:

$$X_c = \prod_{l=0}^{t-1} (C_l)^{c^l} \tag{4}$$

Then, $\mathcal{N}_c$ computes :

$$R_c = e(X_c, pk_c), 1 \leqslant c \leqslant n$$

and employs the following approach for batch verification:

$$\prod_{c=1}^{n} R_c = e(g_2, \prod_{c=1}^{n} Y_c) \tag{5}$$

If the above equation holds, $\mathcal{N}_c$ acknowledges that all the received polynomial commitments $\langle Y_c \rangle_{c=1\sim n}$ are correct and stores the corresponding $\{\pi_i, Y_c\}$.

Each $\mathcal{N}_c$ further uses its own private key sk_c to recover $share_c$ from Y_c.

$$share_c = (Y_c)^{\frac{1}{sk_c}}$$

In the subsequent transmission of $share_c$ by $\mathcal{N}_c$, to ensure that the recipients indeed receive $share_c$ from the corresponding Y_c, N_c needs to provide relevant proof information. First, $\mathcal{N}_c$ selects a random number $r_c \in Z_q^*$ and calculates $B_{c,1} = (share_c)^{r_c}, B_{c,2} = (g_1)^{r_c}$. Then, $\mathcal{N}_c$ calculates:

$$e_c = H_2(share_c \parallel g_1 \parallel Y_c \parallel pk_c \parallel B_{1,c} \parallel B_{2,c}), b_c = r_c + e_c sk_c$$

and finally generates share information $\{share_c, e_c, b_c\}$ that can be used to trace identities and recover keys.

3.10 Traceability of Malicious Nodes

The $\mathcal{M}$ initiates an on-chain request to record user's traceability information $\{\pi_i, PID_i\}$, and the consensus group can search for this information to trace the identity of any malicious user. The smart contract is used to automatically trace the malicious node. if a user has malicious behavior, once the number of $\mathcal{N}_c$ that considers the user to be malicious exceeds the threshold t, the tracing process will be automatically triggered. The specific tracking steps are as follows:

Algorithm 1. Tracking of Malicious Nodes

Require: $\pi_i, [share_1, share_2, \ldots, share_t], [e_1, e_2, \ldots, e_t], [b_1, b_2, \ldots, b_t]$
Ensure: $\pi_i \oplus H_1(S_{1,i})$

for $c \leftarrow 1$ **to** t **do**
  Compute:
  $temp \leftarrow share_c \parallel g_1 \parallel Y_c \parallel pk_c \parallel (share_c)^{b_c} (Y_c)^{-e_c} \parallel {g_1}^{b_c} (Y_c)^{-e_c}$
  ${e_c}^* \leftarrow H_2(temp)$
  if ${e_c}^* \neq e_c$ **then**
    fail
  end if
end for
$S_{1,c} \leftarrow 1$
for $c \leftarrow 1$ **to** t **do**
  $L_c \leftarrow 1$
  for $j \in [1, t]$ **do**
    if $j \neq c$ **then**
      $L_c \leftarrow L_c \times \frac{j}{j-c}$
    end if
  end for
  $S_{1,c} \leftarrow S_{1,c} \times pow(share_c, L_c)$
end for

Each $\mathcal{N}_c$ submits its tracing shares $\{share_c, e_c, b_c\}$ to the smart contract. When the number of tracing shares inputted into the smart contract reaches t, the smart contract automatically executes the tracing algorithm as shown in Algorithm 1 to recover the user's $S_{1,i}$. Eventually, the consensus group will reveal the real identity information $Info_i$ of user $\mathcal{U}_i$ by the following formula:

$$Info_i = \pi_i \oplus H_1(S_{1,i})$$

3.11 Key Recovery

The smart contract is used to automatically recover a user's lost private key. When a user $\mathcal{U}_i$ loses his private key sk_i, the user immediately broadcasts a message indicating the key loss along with important parameter information $\{PID_i, \sigma_i\}$ to the TCB. Each $\mathcal{N}_c$ verifies his identity by the formula (1). If the identity verification of $\mathcal{U}_i$ passes, $\mathcal{N}_c$ can confirm that PID_i represents the user $\mathcal{U}_i$. Then, $\mathcal{N}_c$ responds to the user by submitting $\{share_c, e_c, b_c\}$ to the smart contract. In the same way as the verification process of tracing the identities of malicious nodes, the smart contract automatically performs information verification to ensure that the received $share_c$ comes from the original Y_c. Simultaneously, $\mathcal{U}_i$ submits $\{A_i, sig_{A,i}\}$ to the smart contract, which automatically performs the $\mathcal{F}.SignVerif$ algorithm to verify A_i.

When the identity verification of $\mathcal{U}_i$ and the verification of A_i have both passed, and the number of $\mathcal{N}_c$ responding to $\mathcal{U}_i$ has reached the threshold t, the key recovery procedure of the smart contract will be executed as follows:

$$L_c = \prod_{j=1, j\neq c}^{t} \frac{j}{j-c}, S_{1,i} = \prod_{c=1}^{t} (share_c)^{L_c}, S_{2,i} = \frac{S_{1,i}}{{g_1}^{a_i}} \tag{6}$$

Subsequently, the smart contract only sends $S_{2,i}$ to $\mathcal{U}_i$, and $\mathcal{U}_i$ can finally recover his private key $sk_i = H_3(S_{2,i})$. Any consensus node cannot have access to the related information of A_i, the polynomial commitments of other consensus nodes, and $S_{2,i}$.

4 Scheme Analysis

4.1 Correctness Analysis

Theorem 1. *The UVIS scheme proposed in this paper is correct.*

Proof. To prove the correctness of the proposed UVIS scheme, it only needs to demonstrate equations in (1)–(6) hold.

(1) For the identity information $\{PID_i, \sigma_i\}$, if the computed δ^* is equal to δ_i, then we have:

$${g_1}^{\sigma_i} = {g_1}^{(z_i + \delta_i sk_{mn})} = (pk_{mn})^{\delta^*} \times PID_i$$

Therefore, the Eq. (1) holds.

(2) The ciphertext $\{D, C_1, C_2, C_3\}$ satisfies the Eq. (2).

$$H_4(PID_i)^{C_3} = H_4(PID_i)^{(u+sk_i H_7(D,C_1,C_2))} = C_2 \times pk_i^{H_7(D,C_1,C_2)}$$

(3) The $rk_1 = H_8({d_j}^{k_i} \parallel {K_j}^{\sigma_i} \parallel PID_i \parallel PID_j \parallel pk_i \parallel pk_j)$ used by $\mathcal{U}_i$ for encryption satisfies the Eq. (3).

$$\begin{aligned} rk_1 &= H_8({d_j}^{k_i} \parallel {K_j}^{\sigma_i} \parallel PID_i \parallel PID_j \parallel pk_i \parallel pk_j) \\ &= H_8((PID_j \times (pk_{mn})^{H_3(PID_j)})^{k_i} \parallel ({g_1}^{k_j})^{\sigma_i} \parallel PID_i \parallel PID_j \parallel pk_i \parallel pk_j) \\ &= H_8({K_i}^{z_j + sk_{mn} H_3(PID_j)} \parallel {g_1}^{k_j(z_i + sk_{mn} H_3(PID_i))} \parallel PID_i \parallel PID_j \parallel pk_i \parallel pk_j) \\ &= H_8({K_i}^{\sigma_j} \parallel {d_i}^{k_j} \parallel PID_i \parallel PID_j \parallel pk_i \parallel pk_j) \end{aligned}$$

(4) For the commitments $\langle C_l \rangle_{l=0 \sim t-1}$ published by $\mathcal{U}_i$, if the $f_i(c)$ hidden in X_c is generated by the polynomial $f_i(x)$, then we have:

$$\prod_{l=0}^{t-1} (C_l)^{c^l} = \prod_{l=0}^{t-1} (g_2)^{a_{l,i} c^l} = {g_2}^{\sum_{j=0}^{t-1} a_{l,i} c^l} = {g_2}^{f_i(c)} = X_c$$

Therefore, the Eq. (4) holds.

(5) For the public information $\{\langle X_c \rangle_{c=1 \sim n}, \langle R_c \rangle_{c=1 \sim n}\}$, if $\langle Y_c \rangle_{c=1 \sim n}$ are correct, then we have:

$$\begin{aligned} \prod_{c=1}^{n} R_c &= \prod_{c=1}^{n} e(X_c, pk_c) = \prod_{c=1}^{n} e({g_2}^{f_i(c)}, pk_c) = \prod_{c=1}^{n} e(g_2, Y_i) \\ &= e(g_2, Y_1 \cdot Y_2 \cdots Y_n) = e(g_2, \prod_{c=1}^{n} Y_c) \end{aligned}$$

Therefore, the Eq. (5) holds.

(6) The t or more correct copies of $share_c$ and $S_{1,i}$ satisfy the Eq. (6).

$$\begin{aligned} \prod_{c=1}^{t} (share_c)^{L_c} &= \prod_{c=1}^{t} ((Y_c)^{\frac{1}{sk_c}})^{L_c} = \prod_{c=1}^{t} ({g_1}^{f_i(c)})^{L_c} = {g_1}^{\sum_{c=1}^{t} (f_i(c) \prod_{j=1, j \neq c}^{t} \frac{j}{j-c})} \\ &= {g_1}^{f_i(0)} = S_{1,i} \end{aligned}$$

4.2 Security Analysis

Theorem 2. *If the DL and CDH assumptions hold, identity privacy-preserving mechanism with anonymity and traceability is secure in the distributed environment.*

Proof. In this identity privacy protection mechanism, there are three ways in which an attacker can obtain the user's real identity information:

1. The adversary manages to reveal $share_c$ by exploiting the public information $\{g_1, g_2, X_c, Y_c, pk_c\}$. Once obtaining t copies of $share_c$, the adversary proceeds to recover $S_{1,i}$ and compute $Info_i$ by π_i.

To simplify the proof, let $g_1 = {g_2}^{\alpha}, X_c = {g_2}^{f_i(c)} = {g_2}^{\beta}, pk_c = {g_1}^{sk_c} = {g_2}^{\alpha sk_c} = {g_2}^{\gamma}$, and then we have $Y_c = {pk_c}^{f_i(c)} = {g_2}^{\beta\gamma}$. The adversary tries to obtain $share_c = {g_1}^{f_i(c)} = {g_2}^{\alpha f_i(c)} = {g_2}^{\alpha\beta}$. Therefore, the problem can be transformed into calculating ${g_2}^{\alpha\beta}$, given ${g_2}^{\beta}, {g_2}^{\gamma}$ and ${g_2}^{\beta\gamma}$, for any $\alpha, \beta, \gamma \in Z_q^*$. An adversary, with all publicly available information, might try to calculate ${g_2}^{\alpha\beta}$ from two perspectives:

(1) The adversary may attempt to compute ${g_2}^{\alpha\beta}$ directly from ${g_2}^{\alpha}$ and ${g_2}^{\beta}$. However, according to the CDH assumption, given $g_2, {g_2}^{\alpha}$ and ${g_2}^{\beta}$, for any $\alpha, \beta \in Z_q^*$, there does not exist a probabilistic polynomial-time adversary that can compute ${g_2}^{\alpha\beta}$ with non-negligible advantage. Therefore, this approach contradicts the CDH assumption.
(2) The adversary may consider deriving β from ${g_2}^{\gamma}$ and ${g_2}^{\beta\gamma}$. However, under the DL assumption, given ${g_2}^{\gamma}$ and ${g_2}^{\beta\gamma}$, for any $\gamma \in Z_q^*$, there does not exist a probabilistic polynomial-time adversary that can compute β with non-negligible advantage. Therefore, unless the adversary can obtain β, it is not possible to further compute ${g_2}^{\alpha\beta}$.

In summary, under the DL and CDH assumptions, the adversary is unable to obtain $share_c$ simply by the public information. As a result, the adversary cannot collect enough shares to recover $S_{1,i}$ and further compute $Info_i$.

2. The adversary can destroy up to $t-1$ N_c, then obtain their $\langle sk_c \rangle_{c=1\sim t-1}$ and $\langle share_c \rangle_{c=1\sim t-1}$. The adversary may attempt to recover $S_{1,i}$ and $Info_i$ of U_i through $\{\langle sk_c \rangle_{c=1\sim t-1}, \langle share_c \rangle_{c=1\sim t-1}\}$ and other public information $\{\langle X_c \rangle_{c=1\sim n}, \langle Y_c \rangle_{c=1\sim n}, \langle pk_c \rangle_{c=1\sim n}, \langle C_l \rangle_{l=0\sim t-1}\}$.

Let $g_1 = {g_2}^{\alpha}$ and $C_0 = {g_2}^{a_{0,i}} = {g_2}^{s_{1,i}} = {g_2}^{\beta}$, then the adversary's target is to compute β or ${g_2}^{\alpha\beta}$. Given the information available, the adversary may try to compute β or ${g_2}^{\alpha\beta}$ from the following three perspectives:

(1) The adversary may consider computing β from $C_0 = {g_2}^{\beta}$, which is equivalent to solving the DL problem. Hence, this contradicts the DL assumption.
(2) The adversary may consider computing ${g_2}^{\alpha\beta}$ from ${g_2}^{\alpha}$ and $C_0 = {g_2}^{\beta}$. However, this is equivalent to solving the CDH problem, which contradicts the CDH assumption.
(3) The adversary may consider recovering ${g_2}^{\alpha\beta}$ by $\langle share_c \rangle_{c=1\sim t-1}$. However, according to the Lagrange interpolation theorem, the adversary can recover ${g_2}^{\alpha\beta}$ only if t or more copies of $share_c$ are collected.

$$\prod_{c=1}^{t} (share_c)^{L_c} = \prod_{c=1}^{t} ({g_1}^{f_i(c)})^{L_c} = {g_1}^{\sum_{c=1}^{t} f_i(c) L_c} = {g_1}^{f_i(0)} = {g_2}^{s_{1,i}} = {g_2}^{\alpha\beta}$$

$$L_c = \prod_{j=1, j\neq c}^{t} \frac{j}{j-c}$$

But the adversary possesses at most $t-1$ copies of $share_c$, from which he cannot recover $s_{1,i}$ or $S_{1,i}$. Furthermore, the adversary and the corrupted N_c may try to use the information they possess to break the remaining $\langle share_c \rangle_{c=t\sim n}$ in order to meet the threshold t. However, under the DL and CDH assumptions, we have proved that an adversary cannot obtain shares of the uncorrupted $\mathcal{N}_c$ through the public information.

In conclusion, if the DL and CDH assumptions hold, the adversary cannot reconstruct $S_{1,i}$ and $Info_i$ by corrupting $t-1$ or less $\mathcal{N}_c$.

3. The attacker attempts to violently crack the hidde $Info_i$ by exploiting the public identity protection information π_i.

$$\prod_{c=1}^{t} (share_c)^{L_c} = S_{1,i}, Info_i = \pi_i \oplus H_1(S_{1,i})$$

To protect the real identity of $\mathcal{U}_i$, $\mathcal{M}$ generates the identity protection information $\pi_i = H_1(S_{1,i}) \oplus Info_i$ for $\mathcal{U}_i$. According to the previous text, no adversary can recover $S_{1,i}$. At the same time, as long as the security parameter κ is strong enough, it is difficult to violently crack the real identity information $Info_i$ from the protection information π_i. Therefore, under reasonable security intensity, the adversary cannot disclose $Info_i$ from π_i.

All in all, the proposed identity privacy protection mechanism with anonymity and traceability is secure. The user's identity will be revealed only if there are t or more consensus nodes that perceive the user's malicious behavior.

Theorem 3. *This scheme ensures the confidentiality of data.*

Proof. The proposed proxy re-encryption algorithm serves as the core component of data sharing in UVIS, ensuring the confidentiality of traffic data transmission. When a data owner grants access to a data requester, a re-encryption key is generated for the requester and sent to the proxy node $\mathcal{N}_{\mathscr{L}}$. Then, $\mathcal{N}_{\mathscr{L}}$ uses the re-encryption key to re-encrypt the data ciphertext so that the data requester can decrypt re-encrypted ciphertext to obtain target data by his private key. Throughout this process, the proxy node is only responsible for receiving the re-encryption key from $\mathcal{U}_i$ and re-encrypting the ciphertext, and unable to obtain any useful information from the original or re-encrypted ciphertext. Hence, this scheme achieves data confidentiality.

Theorem 4. *This scheme is capable of resisting collusion attacks.*

Proof. In this scheme, the consensus mechanism selects the leader $\mathcal{N}_{\mathscr{L}}$ from $\mathcal{N}_c$ based on the formula $\mathscr{L} = (num \mod n) + 1$. The leader publishes a random number u for data encryption. Suppose that $Leader_1$ records the random number $u^{(1)}$ generated in this phase in a block with a height of H_1, and also records the metadata containing data information m_1 in the same block. If a data requester $\mathcal{U}_j$ wants to retrieve the data associated with that block, he needs to use $u^{(1)}$ and

his private key sk_j to compute $u_j = H_5(u)^{sk_j}$, then initiate an access request to $\mathcal{U}_i$ with the information u_j. If $\mathcal{U}_i$ allows $\mathcal{U}_j$ to access m_1, $\mathcal{U}_i$ will generate a re-encryption key $rk_{i\to j} = (u_j^{(1)})^{a_i/sk_i}$ using u_j and his private key sk_i, and send it to the proxy node.

After the current consensus ends, the consensus group will elect the next leader $Leader_2$ according to the rules. Similarly, the new random number $u^{(2)}$ published by $Leader_2$ and the metadata of m_2 are recorded in a block with a height of H_2. In the encryption phase, $\mathcal{U}_i$ encrypts m_2 by the following formulas:

$$
\begin{aligned}
D^{(2)} &= e(H_4(PID_i), (u_i{}^{(2)})^{a_i}) \\
C_1{}^{(2)} &= (m_2 \parallel \omega) \oplus (D^{(2)})^{H_6(PID_i \| pk_i)} \\
C_2{}^{(2)} &= H_4(PID_i)^{(u^{(2)})} \\
C_3{}^{(2)} &= (u^{(2)} + sk_i H_7(D^{(2)} \parallel C_1{}^{(2)} \parallel C_2{}^{(2)})) \bmod q
\end{aligned}
$$

If $\mathcal{U}_j$ colludes with the proxy node to access unauthorized m_2, they can only obtain $(H_5(u^{(1)}))^{a_i/sk_i}$ about the previous random number $u^{(1)}$ and further calculate $e(pk_i, (H_5(u^{(1)}))^{a_i/sk_i}) = e(H_4(PID_i), H_5(u^{(1)}))^{a_i}$. However, it differs from $D^{(2)} = e(H_4(PID_i), H_5(u^{(2)}))^{a_i}$, making it ineffective for obtaining m_2. So is the case for other data. Therefore, this scheme effectively resists the collusion attacks between the proxy node and data requesters.

Theorem 5. *This scheme can securely reconstruct the user's original private key in case a user loses his private key.*

Proof. On the basis of achieving identity privacy protection and traceability, this scheme adds the functionality of key recovery. When a user $\mathcal{U}_i$ loses his private key sk_i, the user only needs to submit A_i and his signature $Sig_{A,i}$ to the smart contract, and the smart contract can automatically execute the key recovery procedure to help $\mathcal{U}_i$ recover his private key sk_i. According to the previous text, we have proved that an adversary cannot steal user's $S_{1,i}$ and further obtain $S_{2,i}$, the adversary cannot obtain critical information A_i for key recovery during this process either. Consequently, even if the number of colluding consensus nodes reaches the threshold t, the private key sk_i of $\mathcal{U}_i$ cannot be recovered.

5 Analysis and Comparison

5.1 Functional Analysis

We compared our scheme with the following proxy re-encryption-based data sharing schemes in terms of functionality and security features (see Table 1). Eltayieb et al. [4] proposed a certificateless proxy re-encryption scheme based on cloud blockchain (CPRCB), providing complete data transparency and auditability in cloud servers. Zeng et al. [25] introduced SS-PRE, a conditional proxy re-encryption technique (C-PRE) for fine-grained access control.

Table 1. Comparison of functional features

Scheme	Confidentiality	Integrity	Identity privacy	Identity tracking	Anti-collusion security	Single point of failure prevention	Decentralized storage	Key recovery
Zeng et al. [25]	✓	✓	×	×	✓	×	×	×
Ge et al. [6]	✓	✓	×	✓	✓	×	×	×
Su et al. [18]	✓	✓	×	×	✓	×	×	×
Eltayieb et al. [4]	✓	✓	×	×	×	✓	✓	×
Ours	✓	✓	✓	✓	✓	✓	✓	✓

Ge et al. [6] proposed an identity-based broadcast proxy re-encryption mechanism for secure data sharing and revocation of access permissions. Su et al. [18] proposed a trusted authorization scheme based on proxy re-encryption for nodes on CloudIoT (PRTA), which can update trusted authorization in CloudIoT.

All the above schemes can realize data confidentiality and integrity. However, they all fail to satisfy the user's desire for protecting identity privacy in data sharing. Our scheme not only protects identity privacy, but also allows for identity tracking when necessary. Furthermore, due to the inherent characteristics of the blockchain, the proposed scheme in [4] can prevent single-point failures and the data stored on the blockchain is traceable, but it cannot ensure the traceability of malicious users while protecting users' identity privacy in data sharing, which is a key security feature of data sharing scenario in UVIS. The schemes in [6,18,25] have risks of single-point failures and data loss but we do not, and they can resist collusion attacks. To ensure safe data sharing, the risk of malicious collusion between a proxy and data applicants needs to be considered. We incorporate appropriate defensive measures into the proxy re-encryption, making the scheme resistant to collusion attacks. Moreover, our scheme has a key recovery function for users who lost their private keys.

5.2 Theoretical Analysis

Table 2 compares the computational complexities of the encryption, re-encryption key generation, re-encryption, self-decryption, and re-decryption of proxy re-encryption in our scheme with those in the comparative literature. We focus on analyzing the most time-consuming operations in these phases, such as exponentiations in the G_1 group and bilinear pairing e. Exp and Pair denote

Table 2. Comparison of Computation Complexity

Scheme	Zeng et al. [25]	Ge et al. [6]	Su et al. [18]	Eltayieb et al. [4]	Ours
Encrypt	2Exp+2Pair	8Exp	4Exp	3Exp+1Pair	3Exp+1Pair
ReKeyGen	2Exp	2Exp	2Exp	5Exp	4Exp
ReEncrypt	Exp+1Pair	2Exp+5Pair	1Exp+2Pair	2Exp+4Pair	4Exp+1Pair
Self-Decrypt	1Pair	3Exp+3Pair	–	–	4Exp+1Pair
ReDecrypt	1Exp+2Pair	2Exp+2Pair	1Exp+3Pair	2Exp+4Pair	8Exp

the time of an exponentiation operation in G_1 and a bilinear pairing operation, respectively.

From the computational complexity results of different stages in Table 2, our scheme has considerable computational performance in the re-encryption and re-decryption. The computational efficiency in the encryption and self-decryption stages is moderate. As for re-encryption key generation, it is not advantageous, but the gap is not large.

5.3 Experimental Analysis

This section presents a comprehensive performance simulation and quantitative analysis of the UVIS. We tested the average time consumption of fifteen experiments for each of the six stages of initialization, encryption, re-encryption key generation, re-encryption, self-decryption, and re-decryption in proxy re-encryption. According to practical application scenarios, we simulated the proposed scheme using the PBC library (https://pkg.go.dev/github.com/Nik-U/pbc) based on the go language. The experiment ran on a Linux operating system, with a quad-core Xeon processor and 16GB memory.

According to Fig. 2, the average time spent on the initialization, encryption, re-encryption key generation, re-encryption, self-decryption, and re-decryption is 11.248 ms, 4.035 ms, 6.026 ms, 5.516 ms, 5.729 ms, and 8.018 ms, respectively. From the average time consumed in each stage, the performance of encryption, re-encryption, and self-decryption in the proxy re-encryption scheme is relatively good. The overhead of the re-encryption key generation and re-decryption is slightly higher, mainly due to the exponentiation operations whose execution time is affected by the size and complexity of the input data, as well as the optimization of the functions in the PBC library. These two stages involve more complex exponentiation operations and more variables to be processed, resulting in the time of execution longer. Overall, the computational cost of this scheme is within a reasonable and manageable range.

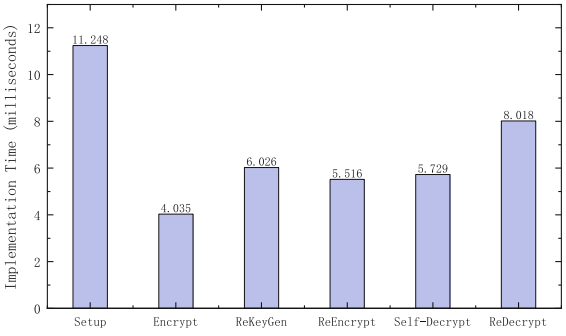

Fig. 2. Time consumed at each stage

6 Conclusion

This paper proposes a UAV-VANET integrated system that leverages UAV to equip VANET with more comprehensive and timely data support. To create a secure and efficient environment for VANET data sharing, UVIS employs a proxy re-encryption data sharing scheme based on the transportation consortium blockchain to achieve precise access control over data. It enables traceability of malicious users while protecting data and identity privacy, effectively resisting collusion attacks caused by proxies and data requesters. Furthermore, UVIS establishes a secure data storage model though the combination of blockchain and IPFS, which effectively solves the issue of secure storage of massive data. Through security analysis, experimental evaluation, and comparison of relevant technologies, UVIS is proved effective and secure as a practical data sharing system.

Acknowledgments. This article is supported in part by the Guangxi Natural Science Foundation under grants AA22068067 and 2023GXNSFAA026236, the National Natural Science Foundation of China under projects 62162017 and 61962012, and the special fund of the High-level Innovation Team and Outstanding Scholar Program for universities of Guangxi.

References

1. Chen, L., Liang, H., Li, X., Ding, Y., Huang, W., Wang, Y., Zhou, X.: Blockchain-based uav-assisted forest supervision and data sharing. In: International Conference on Blockchain and Trustworthy Systems, pp. 251–264. Springer (2022)
2. Da, L., Wang, Y., Ding, Y., Qin, B., Zhou, X., Liang, H., Wang, H.: Cloud-assisted road condition monitoring with privacy protection in vanets. In: 2022 18th International Conference on Mobility, Sensing and Networking (MSN). pp. 304–311 (2022)
3. Da, L., Wang, Y., Ding, Y., Xiong, W., Wang, H., Liang, H.: An efficient certificateless signcryption scheme for secure communication in uav cluster network. In: 2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom), pp. 884–891. IEEE (2021)
4. Eltayieb, N., Sun, L., Wang, K., Li, F.: A certificateless proxy re-encryption scheme for cloud-based blockchain. In: Frontiers in Cyber Security: Second International Conference, FCS 2019, Xi'an, China, November 15–17, 2019, Proceedings 2, pp. 293–307. Springer (2019)
5. Fan, K., Li, F., Yu, H., Yang, Z.: A blockchain-based flexible data auditing scheme for the cloud service. Chin. J. Electron. **30**(6), 1159–1166 (2021)
6. Ge, C., Liu, Z., Xia, J., Fang, L.: Revocable identity-based broadcast proxy re-encryption for data sharing in clouds. IEEE Trans. Dependable Secure Comput. **18**(3), 1214–1226 (2021)
7. Han, X., Tian, D., Zhou, J., Duan, X., Sheng, Z., Leung, V.C.: Privacy-preserving proxy re-encryption with decentralized trust management for mec-empowered vanets. IEEE Trans. Intell. Vehicles, 1–15 (2023)

8. He, J., Ni, Y., Cai, L., Pan, J., Chen, C.: Optimal dropbox deployment algorithm for data dissemination in vehicular networks. IEEE Trans. Mob. Comput. **17**(3), 632–645 (2017)
9. Horng, S.J., Lu, C.C., Zhou, W.: An identity-based and revocable data-sharing scheme in vanets. IEEE Trans. Veh. Technol. **69**(12), 15933–15946 (2020)
10. Kumar, R., Marchang, N., Tripathi, R.: Distributed off-chain storage of patient diagnostic reports in healthcare system using ipfs and blockchain. In: 2020 International Conference on COMmunication Systems & NETworkS (COMSNETS). pp. 1–5 (2020)
11. Li, H., Pei, L., Liao, D., Chen, S., Zhang, M., Xu, D.: Fadb: a fine-grained access control scheme for vanet data based on blockchain. IEEE Access **8**, 85190–85203 (2020)
12. Li, H., Pei, L., Liao, D., Sun, G., Xu, D.: Blockchain meets vanet: an architecture for identity and location privacy protection in vanet. Peer-to-Peer Networking Appl. **12**, 1178–1193 (2019)
13. Li, J., Wang, Y., Ding, Y., Wu, W., Li, C., Wang, H.: A certificateless pairing-free authentication scheme for unmanned aerial vehicle networks. Secur. Commun. Networks **2021**, 9463606 (2021)
14. Liu, J., Jiang, W., Sun, R., Bashir, A.K., Alshehri, M.D., Hua, Q., Yu, K.: Conditional anonymous remote healthcare data sharing over blockchain. IEEE J. Biomed. Health Inform. **27**(5), 2231–2242 (2023)
15. Liu, X., Chen, W., Xia, Y.: Security-aware information dissemination with fine-grained access control in cooperative multi-rsu of vanets. IEEE Trans. Intell. Transp. Syst. **23**(3), 2170–2179 (2020)
16. Noh, S.W., Park, Y., Sur, C., Shin, S.U., Rhee, K.H.: Blockchain-based user-centric records management system. Int. J. Control Autom. **10**(11), 133–144 (2017)
17. Schnorr, C.P.: Efficient identification and signatures for smart cards. In: Advances in Cryptology-CRYPTO 1989 Proceedings 9, pp. 239–252. Springer (1990)
18. Su, M., Zhou, B., Fu, A., Yu, Y., Zhang, G.: Prta: a proxy re-encryption based trusted authorization scheme for nodes on cloudiot. Inf. Sci. **527**, 533–547 (2020)
19. Sun, J., Yao, X., Wang, S., Wu, Y.: Blockchain-based secure storage and access scheme for electronic medical records in ipfs. IEEE Access **8**, 59389–59401 (2020)
20. Wang, Y., Ding, Y., Wu, Q., Wei, Y., Qin, B., Wang, H.: Privacy-preserving cloud-based road condition monitoring with source authentication in vanets. IEEE Trans. Inf. Forensics Secur. **14**(7), 1779–1790 (2019)
21. Wang, Y., Pang, H., Deng, R.H., Ding, Y., Wu, Q., Qin, B., Fan, K.: Secure server-aided data sharing clique with attestation. Inf. Sci. **522**, 80–98 (2020)
22. Wang, Z., Xu, Y., Liu, J., Li, Z., Li, Z., Jia, H., Wang, D.: An efficient data sharing scheme for privacy protection based on blockchain and edge intelligence in 6g-vanet. Wirel. Commun. Mob. Comput. **2022**, 1–18 (2022)
23. Xiong, W., Wang, R., Wang, Y., Wei, Y., Zhou, F., Luo, X.: Improved certificateless aggregate signature scheme against collusion attacks for vanets. IEEE Syst. J. **17**(1), 1098–1109 (2023)
24. Xiong, W., Wang, R., Wang, Y., Zhou, F., Luo, X.: Cppa-d: efficient conditional privacy-preserving authentication scheme with double-insurance in vanets. IEEE Trans. Veh. Technol. **70**(4), 3456–3468 (2021)
25. Zeng, P., Choo, K.K.R.: A new kind of conditional proxy re-encryption for secure cloud storage. IEEE Access **6**, 70017–70024 (2018)
26. Zheng, Q., Li, Y., Chen, P., Dong, X.: An innovative ipfs-based storage model for blockchain. In: 2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI), pp. 704–708 (2018)