Privacy-Preserving Blockchain Supervision with Responsibility Tracking

Baodong Wen[1], Yujue Wang[2], Yong Ding[1,3(✉)], Haibin Zheng[2,4], Hai Liang[1], Changsong Yang[1], and Jinyuan Liu[2]

[1] Guangxi Key Laboratory of Cryptography and Information Security, School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin, China
stone_dingy@126.com
[2] Hangzhou Innovation Institute of Beihang University, Hangzhou, China
[3] Institute of Cyberspace Technology, HKCT Institute for Higher Education, Hong Kong SAR, China
[4] WeBank Institute of Financial Technology, Shenzhen University, Shenzhen, China

Abstract. Blockchain technology is a strategic technology to support the development of digital economy, which helps to promote data sharing, improve the efficiency of communication and build a trusted system. With the continuous development of blockchain technology, security problems caused by lack of regulation are also frequent. Also, the supervision process needs to check the data information on the blockchain, which may lead to the disclosure of users' privacy information. To solve the above problems, this paper proposes a privacy protection blockchain supervision scheme (PBS) that supports ciphertext supervision and malicious user tracking. The PBS supports the supervision of ciphertext data on the blockchain, meaning that regulators do not need to decrypt it and can perform supervision without plaintext. This ensures that the private data of users on the blockchain would not be compromised in the process of supervision. Moreover, the scheme supports the tracking of the sender of the offending information. Theoretical analysis and comparison show that the proposed PBS can effectively ensure the privacy of user data on blockchain, and experimental analysis demonstrates the practicability of the scheme.

Keywords: blockchain supervision · equality test · privacy protection · malicious user tracking

1 Introduction

Blockchain integrates distributed storage, peer-to-peer network, consensus mechanism, cryptography and other technologies, making on-chain data open, transparent, non-tamperable, and traceable. Blockchain has created a new collaboration model that does not rely on a third party to establish trust in an untrusted environment, and has carried out extensive explorations in many fields such as

H. Gao et al. (Eds.): CollaborateCom 2023, LNICST 561, pp. 243–262, 2024.
https://doi.org/10.1007/978-3-031-54521-4_14

finance [3], unmanned aerial vehicle [2], copyright [17] and supply chain [6]. Thus, blockchain builds a brand new value transfer network, which is the key supporting technology for building the next-generation Internet trusted infrastructure.

The difficulty of tampering with blockchain data makes it difficult to modify and delete the data on the blockchain through traditional methods, which increases the difficulty of supervising harmful information on the chain and poses new challenges for information management [20]. At present, there is still a lack of effective supervision methods in the blockchain system. When attackers threaten the security of the system and illegal users use the blockchain to commit illegal acts, the system cannot hold the attackers and illegal users accountable. Therefore, once harmful information such as violence, terrorism and pornography is written into the blockchain, it can not only spread rapidly using its synchronization mechanism, but also difficult to modify and delete these information.

The openness and transparency of blockchain system data is the basis for building a decentralized trust ecosystem, but it also brings serious challenges to the privacy protection of blockchain systems [7]. The open source sharing protocol in the blockchain enables data to be recorded and stored synchronously at all user sides. For attackers, data copies can be obtained in more locations, and useful information such as blockchain applications, users, and network structures can be analyzed. Encrypting the data on the chain is a way to achieve privacy protection [16]. However, in the supervision system, it is difficult for the regulator to effectively review the ciphertext when supervising the data in the blockchain.

Li et al. [5] proposed a two-layer adaptive blockchain supervision model to solve the problems of difficult supervision and poor privacy in the production of off-site modular housing. In this model, the first layer is the user's private side chain, and the second layer is mainly used for communication and transactions between users. The model allows each participant to access the status and records of the entire process, while protecting the privacy of material suppliers, manufacturers and contractors. Wen et al. [15] proposed a blockchain regulatory framework using attribute-based encryption and a dual-chain model. In this architecture, the regulator first obtains plaintext data from the business chain for the first round of supervision. After the supervision is completed, it is encrypted with attribute-based encryption and sent to the regulatory party. The authorized regulatory party can decrypt the ciphertext and conduct the second round of supervision. However, the above scheme cannot supervise the ciphertext data in the blockchain, that is, the regulator must review the plaintext when supervising. Data senders often do not want regulators to obtain their own private information, thus there is a lack of a mechanism to supervise ciphertext in the current blockchain supervision system.

1.1 Our Contributions

In order to solve the problem that it is difficult to supervise the ciphertext data when monitoring the blockchain, this paper proposes a blockchain supervision scheme that supports the equality test of ciphertext. The scheme uses the ciphertext equality test technology to compare the encrypted keywords of

user data and the keywords that central regulatory want to supervise. When the two keywords are the same, the supervision scheme compares the ciphertext of the keywords and outputs the result that the keywords are equal. In order to avoid the problem that a single regulator has a single point of failure, this paper sets up two regulators to perform joint supervision. The central regulatory authorizes two local regulators, and after authorization, the regulators conduct non-interactive supervision, which reduces the efficiency problems caused by interaction. In order to track the illegal users and prevent regulators from forging keywords and abusing tokens, signcryption is used to track accountability. Central regulatory regularly supervise the supervision records, and when they find that there is illegal supervision or improper supervision, they will be held accountable in time.

1.2 Related Works

In terms of blockchain regulation, Marian [8] suggested using unique identifiers to identify users, giving up the anonymity protection provided by the Bitcoin system. This can improve the detection probability of suspicious users, so as to investigate and sanction traders who use the Bitcoin system to conduct illegal and criminal activities. Yong et al. [20] proposed a vaccine regulatory traceability system that combines machine learning with blockchain. The system uses smart contracts to record vaccines, and can also track information about violating vaccines. In order to supervise vaccine production, Peng et al. [9] designed a two-layer blockchain to record and manage production data. The two-layer blockchain stores private information and public information respectively, thus realizing the privacy protection of production information.

Privacy in blockchain is also one of the key research directions. Bünz et al. [1] proposed a distributed payment mechanism for smart contracts, in which account balances are encrypted and cryptographic proofs are used to secure transactions between users. This mechanism is compatible with other smart contract platforms such as Ethereum. Rahulamathavan et al. [10] constructed an IoT application-level privacy protection architecture through attribute-based encryption. This architecture guarantees the confidentiality of data and devices in blockchain-based IoT systems. Zhang et al. [21] proposed a privacy predictor protocol, which extends the data security function of HTTP/TLS protocol to ensure the privacy and non-tampering of data transmission from various private data sources.

Ciphertext equality test allows testing the equivalence of ciphertexts without decrypting them [13]. Depending on the number of servers used for the comparison, it can be classified into single-server model and dual-server model. Under the single-server model, Yang et al. [19] proposed a public key encryption scheme that supports ciphertext equivalence comparison, which can ensure that the tester can detect whether two ciphertexts encrypted by different public keys are equal without decrypting the ciphertext. To address the issue of the lack of ciphertext equivalence comparison in signature and encryption protocols among multiple systems, Xiong et al. [18] proposed a heterogeneous signcryption scheme

for wireless body area network. The scheme aims to meet requirements such as data authenticity, privacy protection, and security, and achieves efficient implementation of the signcryption scheme by addressing critical issues such as key management and certificate management. In order to ensure the secure sharing of data between remote servers, Wang et al. [14] proposed a proxy re-encryption scheme that supports the equivalence comparison of ciphertexts. This scheme supports users to join the group dynamically, and also supports the distinction and proof of encrypted data.

In the single-server model, the ciphertext equivalence comparison process can be performed by a single server. Although this simplifies the system's workflow, it may also be vulnerable to offline keyword guessing attacks, whereby the server constructs keyword ciphertext multiple times and performs equivalence tests using authorization tokens. If the comparison succeeds, it means that the server has successfully guessed the keyword, which violates the principle of not allowing the server to know the keyword and poses a certain security risk. The dual-server model can effectively prevent offline keyword guessing attacks by having two servers collaborate with each other to perform the equivalence comparison process, thereby improving the security and robustness of the system. Tang [11] proposed a ciphertext equivalence comparison scheme that supports user authorization and dual-server interaction. In this scheme, only one authorization instruction is required to allow any authorized party to perform ciphertext equivalence test. In order to protect the privacy of data in cloud servers and resist keyword guessing attacks, Zhao et al. [22] proposed a dual-server authorized ciphertext equivalence test scheme, in which user data is stored in the main server to reduce local storage.

1.3 Organization

The remainder of this paper is organized as follows. Section 2 describes the system model and requirements. A description of our PBS scheme is presented in Sect. 3. In Sect. 4, the security and performance of our PBS scheme are evaluated and compared. Section 5 concludes the paper.

2 System Model and Requirements

2.1 System Model

As shown in Fig. 1, a PBS system consists of three types of entities, namely, central regulatory (CR), local regulatory (LR) and users.

- CR: CR is responsible for registering users and LR. At the same time, CR can grant regulatory authority to LR. If LR violates the supervision rules, CR will track the LR.
- LR: LR is responsible for supervising the ciphertexts of keywords uploaded by user. When LR discovers that there are illegal keywords in user data, LR reports the illegal information to CR.

- User: User uploads data ciphertext to the blockchain. The data ciphertext consists of two parts, the business content ciphertext and the keyword ciphertext. Keywords are used to summarize key parts of the business content.

In the PBS system, users upload the ciphertext of transaction content and keywords to the blockchain. LR must obtain CR's authorization before performing supervision. During the supervision process, two authorized LRs will conduct joint supervision. When illegal data is found, LR will send a violation report to CR, and CR will be responsible for tracking the illegal users. At the same time, CR will supervise the LR's supervision records, and if there is any violation of supervision, LR will be tracked.

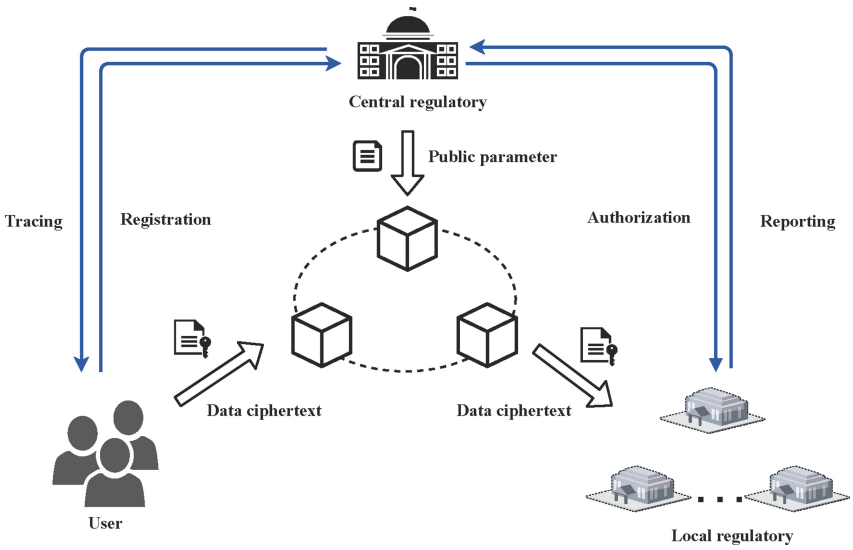

Fig. 1. System model of PBS.

2.2 System Requirements

A secure PBS system has to satisfy the following requirements.

- Privacy protection: Encryption should be used to ensure the privacy of user transaction data and keywords. At the same time, the keywords set by the user should allow the authorized regulator to supervise the ciphertext.
- Resistance of offline keyword guessing attack: When supervising the data uploaded by users, the regulators should be jointly supervised, that is, two regulators should jointly perform the supervision in one supervision process. This can avoid the problem of excessive concentration of regulatory power. In addition, the system should be resistant to offline keyword guessing attacks when two regulators are not colluding.

- Authenticity verification: In the PBS system, malicious users may pretend to be others to upload illegal data. Also, malicious regulators may query keywords that are not within the scope of their own supervision. To defend against this attack, it is necessary to verify the origin of data uploaded by users and keywords regulated by regulators.
- Access control: The system should ensure the access control of the data content on the blockchain, and prohibit unauthorized users from operating the data content, that is, the transaction content uploaded by users is only allowed to be viewed by authorized users, and the keyword ciphertext is only allowed to be supervised by authorized regulators.

A correct PBS system has to satisfy the following requirements.

- The regulatory authorization signcryption ciphertext of CR can be successfully verified by LR.
- $LR\ ID_l$ and $LR\ ID_{l'}$ can correctly regulate the same plaintext encrypted with different public keys.
- The CR can correctly decrypt the keyword signcryption ciphertext.

3 Our PBS Construction

Our PBS framework consists of six procedures, namely, setup, key generation, data uploading, authorization, supervision and tracing. The frequently used notations are listed in Table 1.

Table 1. Notations

Notations	Descriptions
l	Security parameter
G	A multiplicative group with prime order q
G_1, G_2, G_T	Bilinear groups with prime order p satisfying $e : G_1 \times G_2 \rightarrow G_T$
g_1, g_2	Generators G_1 and G_2 respectively
H_1, H_2, H_3, H_4	Cryptographic hash functions
$sk_u = (x_u, y_u)$	Private key of user ID_u
$pk_u = (X_u, Y_u)$	Public key of user ID_u
Tr_u	Transactions sent by user ID_u
K_u	Keywords of user ID_u data
σ_u	Keyword signcryption ciphertext of user ID_u transaction Tr_u
σ_c	User ID_u keyword K_u signcryption ciphertext
T_c, $T_c^{'}$	Authorized supervision tokens for $LR\ ID_l$ and $LR\ ID_{l'}$ respectively
K_c	Keywords that CR want to supervise
ρ_l^*, $\rho_l^{'*}$	Equality test parameter
$\omega_l^{'}$	Supervision records
J	Supervision result
t	Timestamp

3.1 Setup

Given the security parameter $l \in Z^+$, CR chooses a multiplicative group G with prime order q, where g is a generator of group G. Next, CR chooses a bilinear map $e : G_1 \times G_2 \rightarrow G_T$, where G_1 and G_2 are multiplicative groups of prime order p, and they have g_1 and g_2 as their generators respectively.

CR selects four collision-resistant hash functions $H_1 : \{0,1\}^* \rightarrow Z_q$, $H_2 : \{0,1\}^* \rightarrow \{0,1\}^k$, $H_3 : \{0,1\}^* \rightarrow G_1$ and $H_4 : \{0,1\}^* \rightarrow Z_p$, where k is a polynomial in l. Finally, CR publishes $(l, q, p, g, g_1, g_2, e, G, G_1, G_2, G_T, H_1, H_2, H_3, H_4)$ as the public parameters of the system. CR chooses a secure ciphertext-policy attribute-based encryption scheme F and sends it to the blockchain.

3.2 Key Generation

User ID_u chooses $x_u \in_R Z_q, y_u \in_R Z_p$, and calculates

$$X_u = g^{x_u}$$

$$Y_u = g_1^{y_u}$$

Finally, user ID_u outputs the public-private key pair (pk_u, sk_u), where $pk_u = (X_u, Y_u)$ and $sk_u = (x_u, y_u)$. Note that CR and LR can respectively generate their public-private key pairs (pk_c, sk_c) and (pk_l, sk_l) in the similar way.

3.3 Data Uploading

The user ID_u uses attribute-based encryption F to encrypt the transaction data in transaction Tr_u and upload it to the blockchain. For the keyword $K_u \in \{0,1\}^k$ of the transaction Tr_u, user ID_u chooses $\alpha_u, \varphi_u \in_R Z_q$, $\beta_u \in_R Z_p$ and calculates the keyword signcryption ciphertext $\sigma_u = (c_{1,u}, c_{2,u}, c_{3,u}, c_{4,u}, U_u, v_u)$ as follows

$$c_{1,u} = g^{\alpha_u}$$

$$c_{2,u} = g_1^{\beta_u}$$

$$c_{3,u} = H_2(X_c^{\alpha_u}) \oplus K_u$$

$$c_{4,u} = Y_c^{\beta_u} \cdot H_3(K_u)$$

$$I_u = g^{\varphi_u}$$

$$U_u = H_1(I_u \| c_{1,u} \| c_{2,u} \| c_{3,u} \| c_{4,u})$$

$$v_u = U_u x_u + \varphi_u \mod q$$

Finally, user ID_u sends the keyword signcryption ciphertext $\sigma_u = (c_{1,u}, c_{2,u}, c_{3,u}, c_{4,u}, U_u, v_u)$ to blockchain.

3.4 Authorization

The CR completes the authorization by sending the authorization signcryption to $LR\ ID_l$ and $LR\ ID_{l'}$. CR chooses $\eta, \lambda, \gamma \in_R Z_p$, $\varphi_T \in_R Z_q$, and calculates

$$T_{1,c} = g_2^{\gamma}$$

$$T_{2,c} = g_2^{y_c\gamma - \eta H_4(X_l)}$$

$$T_{3,c} = g_2^{\lambda H_4(X_{l'})}$$

$$T_c = (T_{1,c}, T_{2,c}, T_{3,c})$$

$$I_T = g^{\varphi_T}$$

$$U_T = H_1(I_T \| T_c)$$

$$v_T = U_T x_c + \varphi_T \mod q$$

Finally, CR sends the authorization signcryption ciphertext $\sigma_T = (T_c, U_T, v_T)$ to $LR\ ID_l$. After receiving the ciphertext σ_T, $LR\ ID_l$ verifies the following equation.

$$H_1(g^{v_T} \cdot X_c^{-U_T} \| T_c) \stackrel{?}{=} U_T \tag{1}$$

If Eq. (1) holds, it shows that σ_T is sent by CR, which can effectively prevent illegal entities from pretending to be authorized by CR.

Also, CR chooses $\varphi_{T'} \in_R Z_q$ and calculates

$$T_{1,c}^{'} = g_2^{\gamma}$$

$$T_{2,c}^{'} = g_2^{y_c\gamma - \lambda H_4(X_{l'})}$$

$$T_{3,c}^{'} = g_2^{\eta H_4(X_l)}$$

$$T_c^{'} = (T_{1,c}^{'}, T_{2,c}^{'}, T_{3,c}^{'})$$

$$I_T^{'} = g^{\varphi_T^{'}}$$

$$U_T^{'} = H_1(I_T^{'} \| T_c^{'})$$

$$v_T^{'} = U_T^{'} x_c + \varphi_T^{'} \mod q$$

Finally, CR sends the authorization signcryption ciphertext $\sigma_T^{'} = (T_c^{'}, U_T^{'}, v_T^{'})$ to $LR\ ID_{l'}$. After receiving the ciphertext $\sigma_T^{'}$, $LR\ ID_{l'}$ verifies the following equation.

$$H_1(g^{v_T^{'}} \cdot X_c^{-U_T^{'}} \| T_c^{'}) \stackrel{?}{=} U_T^{'} \tag{2}$$

If Eq. (2) holds, it implies that σ_T is sent by CR, which can effectively prevent illegal entities from pretending to be authorized by CR.

3.5 Supervision

Distribute Ciphertext to Be Compared. When CR wants to check whether the transaction keywords uploaded by users contain the keyword K_c through $LR\ ID_l$ and $LR\ ID_{l'}$, CR chooses $\alpha_c, \varphi_c \in_R Z_q^*$, $\beta_c \in_R Z_p^*$, and calculates regulatory keyword signcryption ciphertext $\sigma_c = (c_{1,c}, c_{2,c}, c_{3,c}, c_{4,c}, U_c, v_c)$ as follows

$$c_{1,c} = g^{\alpha_c}$$

$$c_{2,c} = g_1^{\beta_c}$$

$$c_{3,c} = H_2(X_c^{\alpha_c}) \oplus K_c$$

$$c_{4,c} = Y_c^{\beta_c} \cdot H_3(K_c)$$

$$I_c = g^{\varphi_c}$$

$$U_c = H_1(I_c \| c_{1,c} \| c_{2,c} \| c_{3,c} \| c_{4,c})$$

$$v_c = U_c x_c + \varphi_c \mod q$$

Finally, CR sends regulatory keyword signcryption ciphertext $\sigma_c = (c_{1,c}, c_{2,c}, c_{3,c}, c_{4,c}, U_c, v_c)$ to $LR\ ID_l$ and $LR\ ID_{l'}$.

After receiving the ciphertext σ_c, the $LR\ ID_l$ uses the following steps to verify the authenticity.

$$H_1(g^{v_c} \cdot X_c^{-U_c} \| c_{1,c} \| c_{2,c} \| c_{3,c} \| c_{4,c}) \stackrel{?}{=} U_c \tag{3}$$

If Eq. (3) is satisfied, it implies that σ_c comes from the CR, so as to prevent illegal entities from impersonating the CR to illegally distribute regulatory keywords.

Joint Supervision. When the user ID_u publishes the transaction Tr_u to the blockchain, the $LR\ ID_l$ obtains the keyword signcryption ciphertext σ_u uploaded by the user ID_u and verifies the following equation.

$$H_1(g^{v_u} \cdot X_u^{-U_u} \| c_{1,u} \| c_{2,u} \| c_{3,u} \| c_{4,u}) \stackrel{?}{=} U_u \tag{4}$$

If the verification is successful, it indicates that the data source is legal. $LR\ ID_l$ adds records $r = (ID_u, X_u, \sigma_u, ID_l)$ to table a of regulatory records,

$$\rho_l = \frac{e(c_{4,u}, g_2^{\gamma})}{e(c_{2,u}, g_2^{y_c\gamma - \eta H_4(X_l)})}$$

$$\mu_l = e(c_{2,c}, g_2^{\lambda H_4(X_{l'})})$$

and sends ρ_l, μ_l and $c_{2,u}$ to $LR\ ID_{l'}$.

$LR\ ID_{l'}$ computes

$$\rho_l^{'} = \frac{e(c_{4,c}, g_2^{\gamma})}{e(c_{2,c}, g_2^{y_c\gamma - \lambda H_4(X_{l'})})}$$

$$\rho_l^* = \frac{\rho_l}{e(c_{2,u}, g_2^{\eta H_4(X_l)})} \tag{5}$$

$$\rho_l^{'*} = \frac{\rho_l^{'}}{\mu_l} \tag{6}$$

$LR\ ID_{l'}$ compares ρ_l^* with $\rho_l^{'*}$. If $\rho_l^* = \rho_l^{'*}$, K_c is contained in the ciphertext c_u of the user's keyword.

$LR\ ID_{l'}$ adds supervision record $\omega_l^{'}$ to regulatory records table b and uploads it to the blockchain.

$$\omega_{l'} = (ID_\omega \| ID_l \| ID_{l'} \| \sigma_c \| \sigma_u \| J \| t)$$

where ID_ω is the identity of $\omega_l^{'}$, J is the supervision result and t is timestamp.

3.6 Tracing

If an illegal keyword is found in σ_c, the $LR\ ID_l$ will be matched with σ_u in the regulatory record ω_l' and σ_u in the record r in table a of the regulatory record. Then $LR\ ID_l$ will report the records ω_l' and r to CR. CR uses the ID_u in r to trace the user who sent the unauthorized data. After successful tracing, CR will handle the violator according to its own regulatory rules.

CR reviews the regulatory records generated by LR at any time. CR first checks whether it can decrypt the ciphertext σ_u using Eq. (7). If it cannot, it means that the user did not use the public key of the CR when uploading the ciphertext. CR queries Table 1 of $LR\ ID_l$ to find the user ID_u by searching for σ_u and then traces the user. CR uses Eq. (8) to decrypt the ciphertext σ_c in the regulatory record.

$$\hat{K_u} = c_{3,u} \oplus H_2(c_{1,u}^{x_c}) \tag{7}$$

$$\hat{K_c} = c_{3,c} \oplus H_2(c_{1,c}^{x_c}) \tag{8}$$

If the decryption is successful, the CR checks whether the $LR\ ID_l$ and $LR\ ID_{l'}$ search for keywords that are not within the scope of their search. Also, CR checks whether the supervision results are correct. If there is non-compliance by $LR\ ID_l$ and $LR\ ID_{l'}$, the CR conducts identification tracking according to supervision records.

Theorem 1. *The above proposed PBS construction is correct.*

Proof. For the correctness of verification by $LR\ ID_l$ on signcryption ciphertexts σ_T from CR, the Eq. (1) holds as follows

$$\begin{aligned} H_1(g^{v_T} \cdot X_c^{-U_T} \| T_c) &= H_1(g^{v_T} \cdot g^{x_c \cdot (-U_T)} \| T_c) \\ &= H_1(g^{U_T x_c + \varphi_T} \cdot g^{x_c \cdot (-U_T)} \| T_c) \\ &= H_1(g^{\varphi_T} \| T_c) \\ &= H_1(I_T \| T_c) \end{aligned}$$

Therefore, the token sent by CR can be correctly verified.

For the correctness of ciphertext supervision, the Eq. (5) and Eq. (6) holds as follows

$$\begin{aligned} \rho_l^* &= \frac{\rho_l}{e(c_{2,u}, g_2^{\eta H_4(X_l)})} \\ &= \frac{e(c_{4,u}, g_2^{\gamma})}{e(c_{2,u}, g_2^{y_c\gamma - \eta H_4(X_l)}) \cdot e(c_{2,u}, g_2^{\eta H_4(X_l)})} \\ &= \frac{e(Y_c^{\beta_u} \cdot H_3(K_u), g_2^{\gamma})}{e(g_1^{\beta_u}, g_2^{y_c\gamma - \eta H_4(X_l)}) \cdot e(g_1^{\beta_u}, g_2^{\eta H_4(X_l)})} \\ &= \frac{e(g_1^{y_c \cdot \beta_u} \cdot H_3(K_u), g_2^{\gamma})}{e(g_1^{\beta_u}, g_2^{y_c\gamma - \eta H_4(X_l)}) \cdot e(g_1^{\beta_u}, g_2^{\eta H_4(X_l)})} \\ &= \frac{e(g_1^{y_c \cdot \beta_u}, g_2^{\gamma}) \cdot e(H_3(K_u), g_2^{\gamma})}{e(g_1^{\beta_u}, g_2^{y_c\gamma - \eta H_4(X_l)}) \cdot e(g_1^{\beta_u}, g_2^{\eta H_4(X_l)})} \\ &= \frac{e(g_1^{y_c \cdot \beta_u}, g_2^{\gamma}) \cdot e(H_3(K_u), g_2^{\gamma})}{e(g_1^{\beta_u}, g_2^{y_c\gamma}) \cdot e(g_1^{\beta_u}, g_2^{-\eta H_4(X_l)}) \cdot e(g_1^{\beta_u}, g_2^{\eta H_4(X_l)})} \\ &= e(H_3(K_u), g_2^{\gamma}) \end{aligned}$$

$$
\begin{aligned}
\rho_l^{'*} &= \frac{\rho_l^{'}}{\mu_l} \\
&= \frac{\rho_l^{'}}{e(c_{2,c}, g_2^{\lambda H_4(X_{l'})})} \\
&= \frac{e(c_{4,c}, g_2^{\gamma})}{e(c_{2,c}, g_2^{y_c\gamma - \lambda H_4(X_{l'})}) \cdot e(c_{2,c}, g_2^{\lambda H_4(X_{l'})})} \\
&= \frac{e(Y_c^{\beta_c} \cdot H_3(K_c), g_2^{\gamma})}{e(g_1^{\beta_c}, g_2^{y_c\gamma - \lambda H_4(X_{l'})}) \cdot e(g_1^{\beta_c}, g_2^{\lambda H_4(X_{l'})})} \\
&= \frac{e(g_1^{y_c \cdot \beta_c} \cdot H_3(K_c), g_2^{\gamma})}{e(g_1^{\beta_c}, g_2^{y_c\gamma - \lambda H_4(X_{l'})}) \cdot e(g_1^{\beta_c}, g_2^{\lambda H_4(X_{l'})})} \\
&= \frac{e(g_1^{y_c \cdot \beta_c}, g_2^{\gamma}) \cdot e(H_3(K_c), g_2^{\gamma})}{e(g_1^{\beta_c}, g_2^{y_c\gamma - \lambda H_4(X_{l'})}) \cdot e(g_1^{\beta_c}, g_2^{\lambda H_4(X_{l'})})} \\
&= \frac{e(g_1^{y_c \cdot \beta_l}, g_2^{\gamma}) \cdot e(H_3(K_c), g_2^{\gamma})}{e(g_1^{\beta_c}, g_2^{-\lambda H_4(X_{l'})}) \cdot e(g_1^{\beta_c}, g_2^{y_c\gamma}) \cdot e(g_1^{\beta_c}, g_2^{\lambda H_4(X_{l'})})} \\
&= e(H_3(K_c), g_2^{\gamma})
\end{aligned}
$$

Therefore, if K_l and K_u are equal, then ρ_l^* and $\rho_l^{'*}$ are also equal.

For the correctness of the CR's decryption of the keyword signcryption ciphertext, the Eq. (7) holds as follows

$$
\begin{aligned}
\hat{K_u} &= c_{3,u} \oplus H_2(c_{1,u}^{x_c}) \\
&= H_2(X_c^{\alpha_u}) \oplus K_u \oplus H_2(g^{\alpha_u \cdot x_c}) \\
&= H_2(g^{x_c \cdot \alpha_u}) \oplus K_u \oplus H_2(g^{\alpha_u \cdot x_c}) \\
&= K_u
\end{aligned}
$$

Therefore, CR can correctly decrypt ciphertext to get keyword.

4 Analysis

4.1 Security Analysis

Theorem 2. *The PBS scheme proposed in this paper supports privacy protection of user data.*

Proof. When the sender wants to upload data to the blockchain, they will process the data information using ciphertext-policy attribute-based encryption. In addition, for the convenience of regulation, the sender also needs to summarize the keywords of the data to be uploaded. For these keywords, the sender uses encryption algorithms based on equality test techniques to encrypt them. When generating the keyword signcryption ciphertext $\sigma_u = (c_{1,u}, c_{2,u}, c_{3,u}, c_{4,u}, U_u, v_u)$, the

generation process of element $c_{1,u}, c_{2,u}, c_{3,u}, c_{4,u}$ are similar to the algorithm for generating ciphertext in the scheme [11]. The difference is that in the scheme [11], $c_{3,u}$ is concatenated with a random number, and there is an additional ciphertext element $c_{5,u}$. Under the CDH and DDH assumptions, according to Theorem 3 in the scheme [11], the PBS scheme has indistinguishability under chosen ciphertext attack. In addition, LR has no right to decrypt the keyword signcryption ciphertext during regulation, thus the PBS scheme realizes the privacy protection of user data.

Theorem 3. *The PBS scheme proposed in this paper is resistant to offline keyword guessing attacks.*

Proof. When CR wants to regulate user data, it authorizes $LR\ ID_l$ and $LR\ ID_{l'}$ to carry out the specific regulatory process through token. $LR\ ID_l$ generates intermediate parameters from the token and keyword signcryption ciphertext and sends them to $LR\ ID_{l'}$. $LR\ ID_{l'}$ uses the intermediate parameters, token, and keyword signcryption ciphertext to perform ciphertext equality test and outputs the comparison result. According to scheme [11], the regulatory process is jointly executed by two local regulatory agencies. Under the premise that the two local regulatory agencies do not collude, it can prevent a single local regulatory agency from forging keyword signature ciphertext. Therefore, when the two LRs do not collude, the PBS scheme can resist offline keyword guessing attacks.

Theorem 4. *The PBS scheme proposed in this paper supports the verification of the authenticity of data sources.*

Proof. When data sender sends keyword data, they use signcryption technology to process keywords. If regulators discover illegal data, they use the signature of the data sender for positioning and tracking of the violator. To prevent LR from forging authorization tokens and keyword signature ciphertexts, the signcryption technology is also used by the CR when sending tokens and distributing keyword signcryption ciphertexts for regulation, ensuring the authenticity verification of the data source.

Theorem 5. *The proposed PBS scheme supports data access control.*

Proof. The data sender uses ciphertext-policy attribute-based encryption to process the plaintext data and embeds the user data access policy when encrypting the plaintext data. Only authorized users whose attribute set matches the access policy can successfully decrypt the ciphertext to obtain plaintext data when accessing the data. Therefore, only authorized users are allowed to view plaintext data, achieving data access control. In addition, during the supervision period, only LR authorized by CR is allowed to carry out supervision. According to Theorem 4 in [11], under the CDH and DBDH assumptions, the CR can distribute authorizations, i.e., determine the LR*s* that have the authority to regulate. The authorization information can be verified during regulation to prevent unauthorized access. Therefore, the PBS scheme supports data access control.

4.2 Theoretical Analysis

This section compares the schemes of Wang et al. [12], Yong et al. [20] and Peng et al. [9] with our PBS scheme. As shown in Table 2, Wang et al.'s scheme [12], Peng et al.'s scheme [9] and our scheme support privacy protection while supervising. Our PBS scheme supports the supervision of data ciphertext on the blockchain, that is, the regulator can complete the supervision without decrypting the ciphertext, which further enhances the privacy of user data. In addition, the scheme of Yong et al. [20] and our scheme support the tracing of malicious users.

Table 2. Functional comparison

Scheme	Privacy protection	Ciphertext supervision	User tracking	Advantage
Wang et al. [12]	✓	\	\	Improve the safety and consensus efficiency of rice supply chain supervision
Yong et al. [20]	\	\	✓	Support the intelligent supervision of vaccine expiration and fraudulent data
Peng et al. [9]	✓	\	\	High supervision efficiency and low memory overhead
PBS	✓	✓	✓	Support non-interactive ciphertext supervision

We compare the computational complexity of Tang's scheme [11], Huang et al.'s scheme [4], and the PBS scheme. Table 3 lists the theoretical calculation time of the three schemes in key generation, encryption, authorization, ciphertext equality test, and decryption phases, respectively, where T_{exp} represents the time for an exponentiation, T_{PA} represents the time for a bilinear pairing, and T_H represents the time for a hash.

In the key generation phase, Tang's scheme [11], Huang et al.'s scheme [4], and the PBS scheme have the same complexity, requiring two exponentiation operations. In the encryption stage, Tang's scheme [11] and Huang et al.'s scheme [4] both use four exponentiation operations. The PBS scheme requires an additional exponentiation operation to sign the data for regulation, but this ensures the authenticity of the data source. In the authorization stage, Tang's scheme [11] and Huang et al.'s scheme [4] require five exponentiation operations. PBS scheme needs four exponentiation operations. Huang et al.'s scheme [4] requires two hash operations, while the PBS scheme requires three hash operation. In the ciphertext equality test stage, Tang's scheme [11] uses six bilinear pairing operations, Huang et al.'s scheme [4] uses six exponentiation operations, two bilinear pairing operations, and three hash operations. The PBS scheme requires six bilinear pairing operations in the ciphertext equality test stage, which takes longer due to the need for two LRs to jointly execute regulation. Tang's scheme [11] and Huang et al.'s scheme [4] both require two exponentiation operations and four exponentiation operations, respectively, in the decryption stage, while the PBS scheme only requires one exponentiation operation.

Table 3. Theoretical comparison

Scheme	Key generation	Encryption	Authorization	Ciphertext equality test	Decryption
Tang [11]	$2T_{exp}$	$4T_{exp} + 3T_H$	$5T_{exp}$	$6T_{PA}$	$2T_{exp} + 2T_H$
Huang et al. [4]	$2T_{exp}$	$4T_{exp} + 2T_H$	$5T_{exp} + 2T_H$	$6T_{exp} + 2T_{PA} + 3T_H$	$4T_{exp} + 2T_H$
PBS	$2T_{exp}$	$5T_{exp} + 3T_H$	$4T_{exp} + 3T_H$	$6T_{PA}$	T_{exp}

4.3 Experimental Analysis

The PBS scheme was implemented using the Java and Solidity programming languages on a Windows 10 operating system with an Intel(R) Core(TM) i5-7500 CPU @ 3.40 GHz and 16 GB of RAM. The PBS scheme used FISCO BCOS 2.0 as the underlying framework for the consortium blockchain. The Type A pairing was used with a prime order q of 256-bits, and the element size of G was 512-bits. A prime order p of 170-bits was used for the multiplicative group.

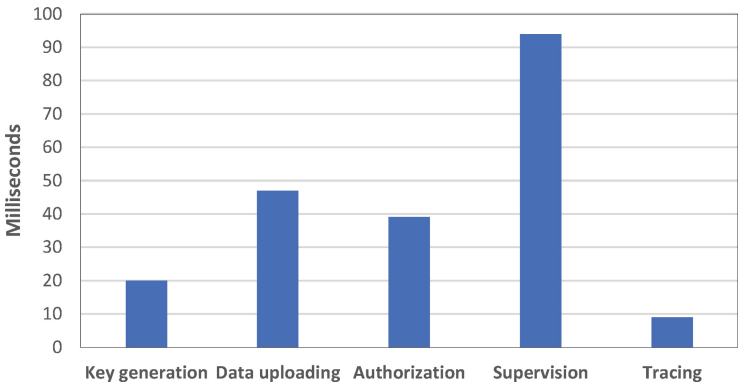

Fig. 2. Time cost of each phase.

Figure 2 shows the time costs of the PBS scheme in the key generation, data uploading, authorization, supervision, and tracing phases. As can be seen from the figure, it takes about 20 ms to generate a public-private key pair for a single user, and about 47 ms to upload keywords, which requires 5 exponential calculations. It takes about 39 ms to authorize a single supervisor. In the supervision phase, there are two steps: setting supervision keywords and joint supervision. Because the setting supervision keywords phase requires 5 exponential calculations, and joint supervision requires 6 bilinear pairing operations, the time cost is relatively large, taking about 94 ms in total. It takes about 9 ms to trace a user.

As shown in Fig. 3, in the keyword encryption stage of PBS scheme, the time required for encryption is counted when the number of keywords increases from 1 to 10. As the number of keywords gradually increases, the required time also

increases. When the number of keywords is 10, the required time is approximately 463.14 ms. Thus, it can be seen from Fig. 3 that the time required for encryption keys is proportional to the number of keywords.

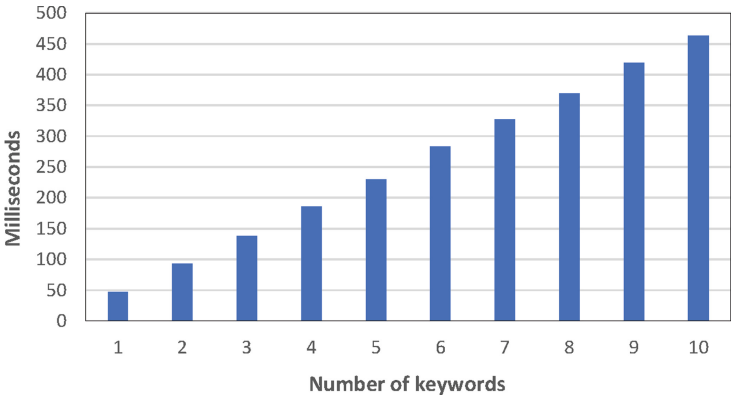

Fig. 3. Time cost of keyword encryption.

During the data upload stage, keyword encryption and ciphertext upload need to be performed. Because it takes a long time to upload the data to the blockchain, the time for uploading the keyword signcryption to the blockchain is listed separately in Fig. 4. The experimental results show that when the number of signcryption ciphertexts is 1, the time required to upload to the blockchain is approximately 0.465 s. As the number of signcryption ciphertexts increases, the time required to upload to the blockchain also increases accordingly. From the observation of the figure, it can be found that when the number of signcryption ciphertexts is 5, the time required for ciphertext upload is approximately 0.491 s. When the number of signcryption ciphertexts is 10, the time required for ciphertext upload is approximately 0.525 s. When the number of signcryption ciphertexts changes from 1 to 10, the time spent uploading to the blockchain increases by approximately 0.06 s. As the number of signcryption ciphertexts increases exponentially, the upload time does not increase exponentially, but shows a linear growth trend.

Figures 5 and 6 show the status information of the signcryption ciphertext uploaded to the blockchain. Figure 5 displays the transaction information on the blockchain, including block hash, block height, contract address, transaction hash and timestamp. The block hash and block height describe the block in which the current transaction is located, while the transaction hash represents the hash value of a transaction in the block, which serves as a unique index for easy retrieval. The required information can be quickly retrieved from the blockchain by using the hash. Figure 6 shows the transaction receipt after it is uploaded to the blockchain, including block hash, gas used, block number and transaction hash. The block hash, block number, and transaction hash in the

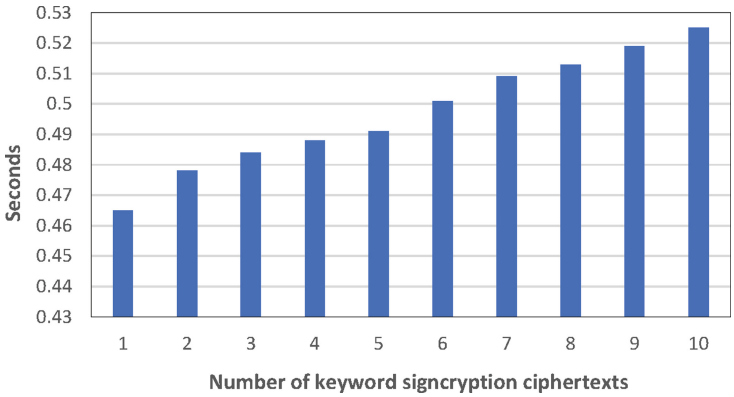

Fig. 4. Time cost of uploading keyword signcryption ciphertext to blockchain.

transaction receipt correspond to the block hash, block height, and transaction hash in the transaction information. The purpose of this correspondence is to better protect and track transaction information. When users or regulatory agencies want to query a transaction, they can quickly locate it by using the block hash, block height, and transaction hash. The gas used represents the amount of gas required to publish this transaction. The more operations required for upload or operation, the more gas is consumed.

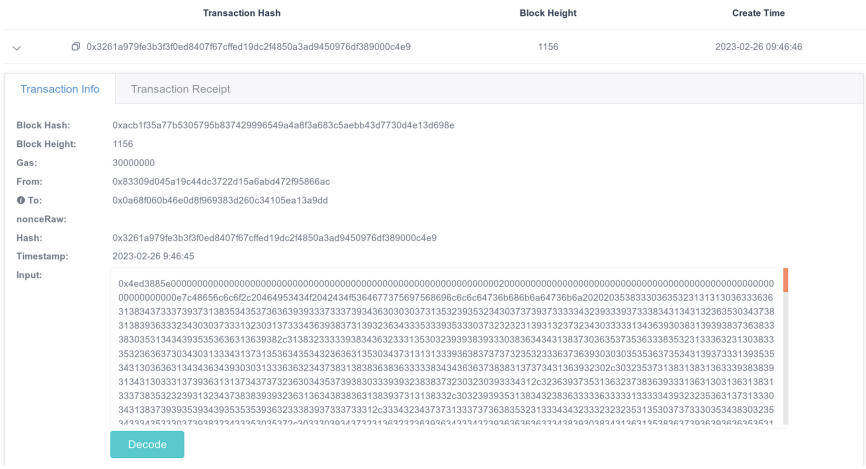

Fig. 5. Transaction information.

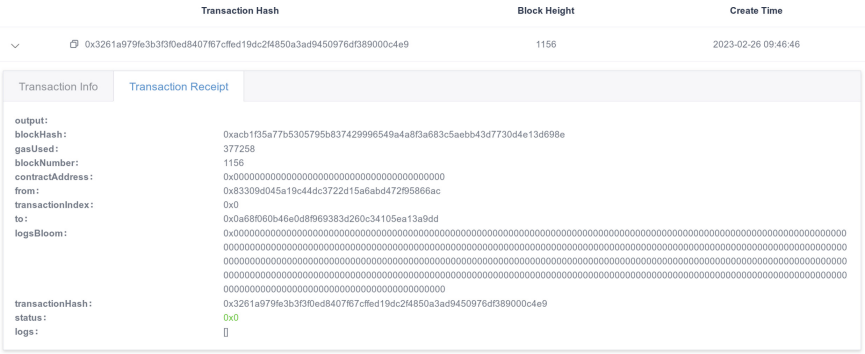

Fig. 6. Transaction receipt.

5 Conclusion

With the rapid development of the digital economy, the crucial role of blockchain as a trust network and trust machine is constantly expanding. However, the data storage structure of blockchain determines that it is difficult to tamper with blockchain data, which also objectively increases the risk of harmful information on the blockchain, as well as the privacy protection of sensitive data. In order to effectively regulate data on blockchain, this paper proposes a blockchain supervision scheme that supports privacy protection, ciphertext supervision and user tracking. The scheme uses ciphertext equality test technology to realize ciphertext supervision on blockchain. This can further protect the privacy of user data. The effectiveness of the scheme is proved by security analysis and experiment analysis.

Acknowledgments. This article is supported in part by the National Key R&D Program of China under project 2022YFB2702901, the Guangxi Natural Science Foundation under grant 2019GXNSFGA245004, the National Natural Science Foundation of China under projects 62162017 and 62172119, Zhejiang Provincial Natural Science Foundation of China under Grant No. LZ23F020012, the Guangdong Key R&D Program under project 2020B0101090002, the Swift Fund Fintech Funding, the special fund of the High-level Innovation Team and Outstanding Scholar Program for universities of Guangxi, and the Innovation Project of GUET Graduate Education 2022YCXS084.

References

1. Bünz, B., Agrawal, S., Zamani, M., Boneh, D.: Zether: towards privacy in a smart contract world. In: Bonneau, J., Heninger, N. (eds.) FC 2020. LNCS, vol. 12059, pp. 423–443. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-51280-4_23
2. Da, L., Liang, H., Ding, Y., Wang, Y., Yang, C., Wang, H.: Blockchain-based data acquisition with privacy protection in UAV cluster network. CMES-Comput. Model. Eng. Sci. **137**(1), 879–902 (2023)

3. Gorkhali, A., Chowdhury, R.: Blockchain and the evolving financial market: a literature review. J. Ind. Integr. Manage. **7**(01), 47–81 (2022)
4. Huang, K., Tso, R., Chen, Y.C., Rahman, S.M.M., Almogren, A., Alamri, A.: PKE-AET: public key encryption with authorized equality test. Comput. J. **58**(10), 2686–2697 (2015)
5. Li, X., Wu, L., Zhao, R., Lu, W., Xue, F.: Two-layer adaptive blockchain-based supervision model for off-site modular housing production. Comput. Ind. **128**, 103437 (2021)
6. Liu, J., Zhang, H., Zhen, L.: Blockchain technology in maritime supply chains: applications, architecture and challenges. Int. J. Prod. Res. **61**(11), 3547–3563 (2023)
7. Ma, Y., Sun, Y., Lei, Y., Qin, N., Lu, J.: A survey of blockchain technology on security, privacy, and trust in crowdsourcing services. World Wide Web **23**(1), 393–419 (2020)
8. Marian, O.: A conceptual framework for the regulation of cryptocurrencies. U. Chi. L. Rev. Dialogue **82**, 53 (2015)
9. Peng, S., et al.: An efficient double-layer blockchain method for vaccine production supervision. IEEE Trans. Nanobiosci. **19**(3), 579–587 (2020)
10. Rahulamathavan, Y., Phan, R.C.W., Rajarajan, M., Misra, S., Kondoz, A.: Privacy-preserving blockchain based IoT ecosystem using attribute-based encryption. In: 2017 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), pp. 1–6. IEEE (2017)
11. Tang, Q.: Public key encryption schemes supporting equality test with authorisation of different granularity. Int. J. Appl. Cryptogr. **2**(4), 304–321 (2012)
12. Wang, J., et al.: Blockchain-based information supervision model for rice supply chains. Comput. Intell. Neurosci. **2022** (2022). Article ID 2914571
13. Wang, Y., Pang, H., Deng, R.H., Ding, Y., Wu, Q., Qin, B.: Securing messaging services through efficient signcryption with designated equality test. Inf. Sci. **490**, 146–165 (2019)
14. Wang, Y., et al.: Secure server-aided data sharing clique with attestation. Inf. Sci. **522**, 80–98 (2020)
15. Wen, B., Wang, Y., Ding, Y., Zheng, H., Liang, H., Wang, H.: A privacy-preserving blockchain supervision framework in the multiparty setting. Wirel. Commun. Mob. Comput. **2021** (2021). Article ID 5236579
16. Wen, B., Wang, Y., Ding, Y., Zheng, H., Qin, B., Yang, C.: Security and privacy protection technologies in securing blockchain applications. Inf. Sci. **645**, 119322 (2023)
17. Xiao, X., Zhang, Y., Zhu, Y., Hu, P., Cao, X.: Fingerchain: copyrighted multi-owner media sharing by introducing asymmetric fingerprinting into blockchain. IEEE Trans. Netw. Serv. Manage. **20**, 2869–2885 (2023)
18. Xiong, H., Hou, Y., Huang, X., Zhao, Y., Chen, C.M.: Heterogeneous signcryption scheme from IBC to PKI with equality test for WBANs. IEEE Syst. J. **16**(2), 2391–2400 (2021)
19. Yang, G., Tan, C.H., Huang, Q., Wong, D.S.: Probabilistic public key encryption with equality test. In: Pieprzyk, J. (ed.) CT-RSA 2010. LNCS, vol. 5985, pp. 119–131. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-11925-5_9
20. Yong, B., Shen, J., Liu, X., Li, F., Chen, H., Zhou, Q.: An intelligent blockchain-based system for safe vaccine supply and supervision. Int. J. Inf. Manage. **52**, 102024 (2020)

21. Zhang, F., Maram, D., Malvai, H., Goldfeder, S., Juels, A.: DECO: liberating web data using decentralized oracles for TLS. In: Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security, pp. 1919–1938 (2020)
22. Zhao, M., Ding, Y., Tang, S., Liang, H., Wang, H.: Public key encryption with authorized equality test on outsourced ciphertexts for cloud-assisted IoT in dual server model. Wirel. Commun. Mob. Comput. **2022** (2022). Article ID 4462134