Hindawi
Security and Communication Networks
Volume 2021, Article ID 7094910, 8 pages
https://doi.org/10.1155/2021/7094910

Research Article

Privacy-Preserving Scheme in the Blockchain Based on Group Signature with Multiple Managers

Fei Tang,[1,2] **Zhuo Feng**,[1] **Qianhong Gong**,[3] **Yonghong Huang**,[2] **and Dong Huang**[4]

[1]*College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China*
[2]*School of Cyber Security and Information Law, Chongqing University of Posts and Telecommunications, Chongqing 400065, China*
[3]*School of Software Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China*
[4]*School of Information Engineering, Chongqing Electromechanical Vocational and Technical University, Chongqing 402760, China*

Correspondence should be addressed to Zhuo Feng; s190231165@stu.cqupt.edu.cn

Received 13 September 2021; Accepted 3 November 2021; Published 24 November 2021

Academic Editor: Youwen Zhu

Group signature can provide the privacy-preserving authentication mechanism for the blockchain. In the traditional blockchain privacy-preserving scheme based on the group signature, there is only one group manager to revoke the anonymity. Thus, the traditional scheme will have single point of failure and key escrow problems. To solve these problems, we propose a privacy-preserving scheme in the blockchain based on the group signature with multiple managers. Our scheme is constructed based on bilinear pairing and the technique of distributed key generation. Finally, we analyze the application of the proposed scheme in the field of blockchain-based provable data possession (PDP), as well as the correctness and security of the scheme.

1. Introduction

Blockchain is the core technology of the system [1]. Blockchain has the characteristics of anonymity, tamper resistance, decentralization, unforgeability, and traceability so that it has attracted extensive attention from the outside world. Moreover, the transaction content on the blockchain is transparent, so all consensus nodes in the blockchain can verify and record this transaction. However, it is due to the transparent characteristics of blockchain ledger that have brought about the problem of user privacy leakage. Research shows that, through a large amount of analysis of these transparent data, it is possible to design a deanonymity scheme, which will lead to the leakage of user privacy. In practical applications, users do not want their transaction information to be placed on the blockchain in a transparent manner. Therefore, how to solve the privacy problem of users on the blockchain is an important challenge.

1.1. Privacy Preservation. The problem that cannot be ignored in blockchain technology is privacy leakage [2]. The privacy preservation of the blockchain includes the anonymity of users and the confidentiality of the content. The privacy-preserving scheme of the blockchain is implemented mainly based on the following three technologies:

(1) Shuffling technology: the purpose of shuffling is to disrupt the correspondence between the input and output so that other users do not know the information of the transaction user, so as to realize the untraceability of transactions. In 1981, Chaum [3] proposed the concept of a shuffling network, but the shuffling protocol requires the participation of trusted authority. Subsequently, in 2014, Bonneau et al. [4] proposed a Mixcoin mechanism with trusted authority. As long as one of the nodes is honest, the privacy of the scheme can be guaranteed.

During the same period, Maxwell [5] proposed the decentralized shuffling protocol named Coinjoin. It places the transactions of multiple users in one bitcoin transaction so that others do not know the relationship between multiple input addresses and output messages. After that, the researchers also proposed CoinShuffle [6] and CoinShuffle^{++} [7] according to the scheme in [5].

(2) Zero-knowledge proof: in order to provide better anonymity, Miers et al. [8] proposed Zerocoin, a digital currency scheme with anonymity based on zero-knowledge proof. Their scheme ensures the nonrelevance of the transaction by hiding the user's address and cutting off the contact between the two parties in this transaction. Subsequently, in 2014, Ben-Sasson et al. [9] proposed a new digital currency scheme Zerocash, which uses a more concise, noninteractive, zero-knowledge proof.

(3) Ring signature: the purpose of ring signature is to hide the real transactions in a collection so that other users do not know the identity of the actual participants. In 2016, Shen and Adam [10] proposed a blockchain secret transaction scheme based on the ring signature. In their scheme, they randomly selected irrelevant addresses and then performed ring signature together with the transaction party to achieve the purpose of confusing the identity of the transaction party. At present, ring signature has been widely used in the blockchain, for example, Monero [11].

1.2. Group Signatures. Group signature is a kind of privacy-preserving authentication scheme which was introduced by Chaum and Van Heyst [12] in 1991. It is widely used in privacy-preserving authentication due to its anonymity. In 2007, Guo et al. [13] proposed a conditional privacy-preserving authentication security framework based on the group signature for vehicle communication networks. Guo et al. mentioned that a security authentication scheme using a group signature can satisfy message integrity, privacy, and traceability. Park et al. [14] proposed distributed key management based on RSU in 2011 to manage group keys, dividing the entire VANET into several subareas, which are managed by the group manager in each area. In addition to having a management entity, RSU is also responsible for managing part of the group key in a distributed manner. In 2012, Sun et al. [15] designed a distributed key management scheme, which divides the entire domain of VANET into several subareas. At the same time, each regional group manager provides distributed key management services for vehicles. This scheme restricts authorization to specific areas and is continuous in time, but the anonymous nature of the group signature makes it possible for malicious users to broadcast forged messages. In 2017, Islam et al. [16] proposed an effective password-based conditional privacy-preserving authentication and group key generation protocol for VANET to provide group key generation, user leave, user join, and password change features. Since the scheme is bilinear-pairing free, it is lightweight in terms of calculation and communication. In 2018, Cui et al. [17] proposed a conditional privacy-preserving authentication scheme based on the hash function, which does not use complex bilinear mapping and elliptic curve encryption to reduce authentication efficiency. At the same time, a group key agreement mechanism based on the Chinese remainder theorem (CRT) is proposed to distribute the group key of authenticated vehicles. When vehicles join and leave the group, the group key can be updated.

In addition, researchers studied the identity-based group signature schemes according to Shamir's concept [18]. An identity-based group signature is a combination of identity-based signature [19] and group signature [12]. Thus, it has the advantages of these two types of signatures. Many schemes have been proposed so far. For example, Cheng et al. [20] constructed an identity-based group signature scheme by using bilinear pairing. Zhang and Ye [21] proposed an identity-based threshold group signature based on the discrete logarithm problem. Ma [22] gave a generic construction of the identity-based group signature. Pulagara and Alphonse [23] proposed an identity-based conditional privacy-preserving authentication method based on elliptic curve cryptography and proposed a group key management scheme. Any vehicle joining or leaving the group will modify the group key to ensure forward security and backward security.

1.3. Our Motivation and Contributions. Group signature can not only protect privacy of transaction participants but also in the event of a transaction dispute. Group manager can open the signature and reveal the true identity of the transaction participants. Thus, group signature has application value in the blockchain, but generally speaking, group manager in the group signature scheme is a single authority so that the group signature will have a single point of failure and key escrow problem.

Therefore, in order to solve the above problems, we propose a privacy-preserving scheme in the blockchain based on the group signature with multiple managers. We use the multiauthority key distribution mechanism to implement the identity-based group signature so that the key generation of group members no longer depends on a single authority. Our scheme can not only realize the privacy preservation of group members but also solve the single point of failure and key escrow problem. In addition, we specifically apply our scheme to the field of blockchain-based provable data possession. Under the multicloud architecture, our scheme realizes the anonymous authentication of the cloud server, which can protect the privacy of the cloud service provider while providing PDP authentication. When the PDP fails to verify, the data owner can apply to find the real signer to protect the interests of the data owner.

1.4. Paper Organization. The rest of the paper is organized as follows. Section 2 introduces some preliminaries including bilinear pairing, blockchain, and definitions. Section 3 presents the scheme of the multimanager group signature.

Section 4 analyzes the security of the proposed scheme. Section 5 gives an application of our scheme. Finally, Section 6 concludes the paper.

2. Preliminaries

2.1. Bilinear Pairings. Let G_1 and G_2 be two cyclic additive groups, respectively, whose orders are a prime p. Let e: $G_1 \times G_1 \longrightarrow G_2$ be a bilinear pairing with the following properties:

(1) Bilinearity: for any $a, b \in Z_p$ and $R, S \in G_1$, the equation $e(R^a, S^b) = e(R, S)^{ab}$ holds

(2) Nondegeneracy: there are $R, S \in G_1$ such that $e(R, S) \neq 1_{G_2}$

(3) Computability: there are effective algorithms to compute the value of $e(R, S)$ for any $R, S \in G_1$

2.2. Blockchain. Blockchain is the underlying technology of Bitcoin [1], which is essentially a distributed database. The blockchain adopts the linked list data structure. The block is composed of the block head and block body. All blocks form a chain structure according to the hash value. Blockchain is a very new network form, which uses cryptography, hash function, and proof of work (Pow). The miners package the legitimate transactions into the "Merkle tree" of the candidate block, fill the hash of the previous block into the new block header, and finally run the consensus mechanism to find the random value suitable for the new block. In summary, in each block head in the blockchain, there is the hash value of the previous block Pre_Hash, the timestamp Timestamp indicating the time when the block was generated, the hash $\text{Hash}_{\text{Root}}$ of the root of the "Merkle tree," and the random value Nonce. The basic structure is shown in Figure 1. With the rapid development of the blockchain, it is also used in many other fields, such as smart grid [24], IoT [25], anonymous authentication [26], and electronic health records [27].

2.3. Provable Data Possession. Storage services are an important part of the cloud computing field. Users store their data in cloud servers, and thus, they can provide a convenient data sharing method. Data stored on the cloud server may be damaged due to external or internal security threats. Therefore, the first provable data possession (PDP) scheme was proposed by Ateniese et al. [28] in 2007. It enables users to know whether the files stored on cloud servers are complete. As time goes on, researchers have proposed some other PDP schemes and their variants based on Ateniese's work, such as [29–31].

A provable data possession scheme includes two different entities, client and cloud server, and its specific protocol is a collection of four polynomial-time algorithms (KeyGen, TagBlock, GenProof, and CheckProof) such that

(1) $\text{Key Gen}(1^k) \longrightarrow (pk, sk)$ is a probabilistic key generation algorithm run by the client. It takes a security parameter k as the input and returns a pair of public and secret keys (pk, sk).

(2) $\text{Tag Block}(pk, sk, m) \longrightarrow T_m$ is an algorithm run by the client. It takes as inputs a public key pk, a secret key sk, and a file block m and returns the verification metadata T_m.

(3) $\text{Gen Proof}(pk, F, \text{chal}, \Sigma) \longrightarrow P$ is run by the cloud server in order to generate a proof of possession. It takes a public key pk, an ordered collection F of blocks, a challenge $chal$, and an ordered collection Σ which is the verification metadata corresponding to the blocks in F as the input and returns a proof of possession P.

(4) $\text{Check Proof}(pk, sk, \text{chal}, P) \longrightarrow 1/0$ is run by the client. It takes as inputs a public key pk, a secret key sk, a challenge $chal$, and a proof of possession P and returns an integer to indicate whether the verification is passed.

2.4. Group Signatures without a Trusted Party. Traditional group signature is limited in some aspects. For example, the downtime of the group manager may lead to the collapse of the whole group; the untrusted group manager may cause the anonymity of group members not to be guaranteed. Thus, we present the system model of the multimanager group signature, which changes the group manager from a single trusted party to multiple trusted parties and realizes the distributed generation of each group member's private key. Our scheme includes two kinds of different entities: group manager and group member.

(1) Group manager: it is an entity consisting of multiple authorities. Its main function is to distribute the private keys of the group members who join this group and find out who is the signer accurately when the group signature needs to be opened.

(2) Group member: it is an entity that has its own public key and private key distributed from the group member. It can sign messages anonymously on behalf of the entire group.

2.4.1. Definitions. We give the formal definition of the multimanager group signature scheme. Subsequently, we present the security requirements that our scheme needs to meet.

Our scheme consists of six algorithms: Setup, Extract, Join, Sign, Verify, and Open. The following is a detailed formal description of the six algorithms:

(1) $\text{Setup}(1^\lambda) \longrightarrow (\text{params}, F_{\text{ID}}, A_{\text{ID}}, mpk, msk)$: it takes a security parameter λ as the input and returns public parameters params, each authority's public and private key pair $(F_{\text{ID}}, A_{\text{ID}})$, and system's master public and secret key pair (mpk, msk)

(2) $\text{Extract}(msk, \text{id}) \longrightarrow (\text{pk}, sk, sk_{\text{id}})$: it takes as inputs the master secret key msk and a group member's

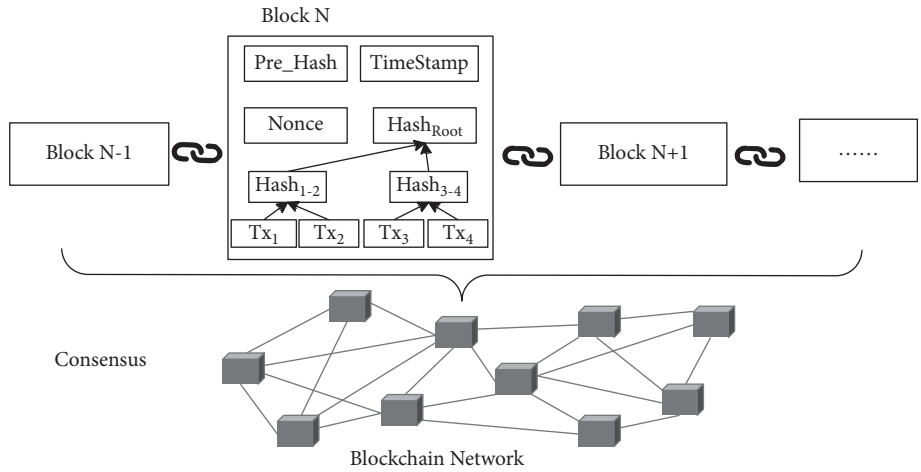

Figure 1: The basic structure of the blockchain.

identity id and returns the group member's public key pk and two secret keys (sk, sk_{id})

(3) Join $(pk, \mathrm{id}, sk_{\mathrm{id}} \longrightarrow \mathrm{cert}_{\mathrm{id}})$: it takes as inputs a group member's public key pk, identity id, and one private key sk_{id} and returns a member certificate $\mathrm{cert}_{\mathrm{id}}$

(4) Sign $(m, sk) \longrightarrow \sigma$: it takes as inputs messages m and one secret key sk and returns a group signature σ

(5) Verify $(pk, \sigma, m) \longrightarrow 1/0$: it takes as inputs a group member's public key pk, a group signature σ, and messages m and returns whether σ is a correct signature of these messages

(6) Open $(msk, \mathrm{cert}_{\mathrm{id}}, \sigma) \longrightarrow \mathrm{id}$: it takes as inputs the master secret key msk, a member certificate $\mathrm{cert}_{\mathrm{id}}$, and a group signature σ and returns the group member's identity id

2.4.2. Security Requirements. A practical multimanager group signature scheme must satisfy the following security requirements:

(1) Correctness: our scheme must be able to complete the verification of the signature. In other words, when the signature is correct, it must be able to pass verification.

(2) Unforgeability: a user who has not registered with the group manager cannot forge the correct group signature even if it can get the public parameters. In short, as long as it is not a member of this group, it is impossible to forge a group signature.

(3) Anonymity: no matter how many times a group member signs, it is impossible for an external member to know who signed it. In the group, the anonymity of group members is guaranteed, except that the group manager can determine the group membership.

(4) Collusion attack prevention: when there are some authorities in the system who want to collusion to leak the data and key of group members, the group members may suffer huge losses, but our scheme can prevent this from happening unless the number of authorities participating in the collusion attack exceeds the threshold.

3. Multimanager Group Signature Scheme

In this section, we consider the multimanager id-based group signature scheme from bilinear pairings, and our scheme consists of six algorithms: Setup, Extract, Join, Sign, Verify, and Open. The detailed description is given below.

3.1. Setup. The setup algorithm consists of two phases.

3.1.1. System Setup. Let G and G_T be two groups with the same big prime order p, and define a bilinear map $e: G \times G \longrightarrow G_T$. Let g be the generator of G. Define the following cryptographic hash functions $H_0: \{0,1\}^* \longrightarrow G$ and $H_1: \{0,1\}^* \longrightarrow Z_p^*$. The system server assigns a different identity ID_i to each authority and defines the number of given authorities as n, and the threshold in key generation is t. Finally, publish the parameters

$$\mathrm{params} = \{G, G_T, p, g, e, H_0, H_1, n, t, \mathrm{ID}_i\}. \tag{1}$$

3.1.2. Authority Setup. For each authority, randomly select $c_i \in Z_p^*$, compute $C_i = g^{c_i}$, and finally send C_i to other authorities. After all authorities receive C_i, they compute $h = \prod_{i=1}^{n} C_i$ and publish it. For each authority, randomly select two polynomials with the order $t-1$ on Z_p^*:

$$\begin{aligned} f_i(x) &= a_{i0} + a_{i1}x + \cdots + a_{i(t-1)}x^{t-1}, \\ f_i'(x) &= b_{i0} + b_{i1}x + \cdots + b_{i(t-1)}x^{t-1}. \end{aligned} \tag{2}$$

After that, each authority calculates and broadcasts $B_{ik} = g^{a_{ik}} h^{b_{ik}}$, where $k = 0, 1, \ldots, n-1$. Each authority takes the identity ID of other authorities to calculate the secret value $s_{ij} = f_i(\mathrm{ID}_j) \bmod p$ and $s_{ij}' = f_i'(\mathrm{ID}_j) \bmod p$, where

$j = 1, 2, \ldots, n$. Then, send them to ID_j, where $j \neq i$. When the authority receives the secret value, it verifies whether the following equation holds:

$$g^{s_{ji}} h^{s_{ji}\prime} = \prod_{k=0}^{n-1} \left(B_{jk}\right)^{ID_i^k}. \tag{3}$$

If the equation holds, ID_i considers ID_j to be the authority of honesty. Otherwise, ID_i requires ID_j to resend the secret value. After authority ID_i receives the secret value $s_{ji}\,(j = 1, \ldots, i-1, i+1, \ldots n)$ from other $n-1$ authorities, it generates its own secret value $F_{\mathrm{ID}_i} = \sum_{j=1}^{n} s_{ji}$ and sets its private key as F_{ID_i}. Correspondingly, the public key of ID_i is $A_{\mathrm{ID}_i} = g^{F_{\mathrm{ID}_i}}$. Then, the system server generates the system main public key y,

$$y = \prod_{i=1}^{n} A_{\mathrm{ID}_i}^{\prod_{j=1, j\neq k}^{t} \mathrm{ID}_j/\mathrm{ID}_j - \mathrm{ID}_k} = g^s, \tag{4}$$

according to the public key of all authorities, where s is the main private key. Ultimately, both the main public key y and the public key A_{ID_i} are public. In the group signature scheme, all authorities work together to act as the group manager (GM).

3.2. Extract. Firstly, the user chooses a random value $x \in Z_p^*$ for signing, then calculates the public key $y_1 = g^x$, and sends (y_1, id) to the GM. Secondly, the user applies to the GM for the secret key. The user applies to authority ID_j to join the system with id_i, and ID_j returns authorization information $S_{ij} = H_0(\mathrm{id}_i)^{F_{\mathrm{ID}_j}}$. After that, id_i sends S_{ij} to other authorities ID_k, where $k \in S, S \subset [1, N], |S| = t$. Then, ID_k verifies whether the equation $e(S_{ij}, g) = e(H_0(\mathrm{id}_{ij}), A_{\mathrm{ID}_j})$ holds. If the equation holds, ID_k sends partial secret key $S_{ik} = H_0(\mathrm{id}_i)^{F_{\mathrm{ID}_k}}$ to id_i. When id_i receives the partial key from ID_k, it verifies whether the equation $e(S_{ik}, g) = e(H_0(\mathrm{id}_i), A_{ID_k})$ holds. After id_i receives and verifies T partial secret keys from ID_k, it calculates its own secret key sk_{id_i} for opening:

$$\mathrm{sk}_{\mathrm{id}_i} = \prod_{k\in S} S_{ik}^{\prod_{k\in S/j} \mathrm{ID}_k/\mathrm{ID}_k - \mathrm{ID}_j}. \tag{5}$$

Finally, the user has two secret keys (x, sk_{id}).

3.3. Join. When a user wants to join this group, it chooses a random value $d \in Z_p^*$, then calculates g^d and g^{xd}, and then sends $\{y_1, g^d, g^{xd}, \mathrm{id}, sk_{\mathrm{id}}\}$ to the GM. The GM verifies whether the equation

$$e\left(g^{xd}, g\right) = e\left(y_1, g^d\right) \tag{6}$$

holds. If the equation holds, the user becomes a member of this group; otherwise, the user fails to join this group. Among them, $(g^{xd}, sk_{\mathrm{id}})$ is defined by the GM as a member certificate of the user.

3.4. Sign. Firstly, the user confirms that it needs to sign the information $m \in \{0,1\}^r$ and chooses an integer $k \in Z_p^*$. Secondly, the user calculates the following values, respectively:

(i) $a \in Z_p^*, u = g^{kxd}, v = g^{ad}$
(ii) $b = H_0(\mathrm{id} \,\|\, u + v), r = b^d$
(iii) $h_1 = H_1(m \,\|\, u + v + r)$
(iv) $w = y_1^{khd} r^x$

Finally, the signature of the message m is (u, v, r, w).

3.5. Verify. After receiving the signature, the verifier can verify the correctness of the signature based on public information. Firstly, the verifier computes $h_1' = H_1(m \,\|\, u + v + r)$ and determines whether it is equal to h_1 and then verifies whether the equation

$$e(w, g) = e\left(g^{h_1\prime}, u\right) e(r, y_1), \tag{7}$$

holds or not. If it holds, output 1; otherwise, output 0.

3.6. Open. If there is a problem with the signature and the verifier wants to know who the signer is, then t managers can cooperate to track the identity of the signer.

$$\begin{gathered} e\left(g^{xd}, g\right) = e\left(y_1, g^d\right), \\ e\left(\prod_{k\in S} S_{ik}^{\prod_{k\in S/j} \mathrm{ID}_k/\mathrm{ID}_k - \mathrm{ID}_j}, g\right) = e(H_0(\mathrm{id}), y). \end{gathered} \tag{8}$$

4. Security Analysis

In this section, we analyze the security of our multimanager group signature scheme.

4.1. Correctness

Theorem 1. *If $e(w, g) = e(g^{h_1'}, u) e(r, y_1)$ is correct, the signature is valid.*

The correctness can be proved by the following equation.

Proof.

$$\begin{aligned} e(w, g) &= e\left(y_1^{khd} r^x, g\right) \\ &= e\left(y_1^{khd}, g\right) e\left(r^x, g\right) \\ &= e\left(g^{xkh_1 d}, g\right) e\left(r, g^x\right) \\ &= e\left(g^{h_1}, g^{xk\,d}\right) e(r, y_1) \\ &= e\left(g^{h_1}, u\right) e(r, y_1) \\ &= e\left(g^{h_1\prime}, u\right) e(r, y_1). \end{aligned} \tag{9}$$

□

4.2. Unforgeability. Because a part of the user's key is generated by the multiple authorities, the detection of the user's part of the key also needs multiple authorities to complete the inspection. In other words, if the user is not a

member of this group, it is absolutely impossible to forge this part of the key. Furthermore, due to the difficulty of discrete logarithm, it is impossible for an invalid user or group manager to find the secret key from the valid user's public key. Therefore, it is also impossible for a user who does not belong to the group to forge signatures, and the group manager cannot forge a legal signature.

4.3. Anonymity. We cannot find any information about the identity of the signer from the group signature (u, v, r, w). In our scheme, every element in the group signature is generated by modular exponentiation, so it is impossible to determine the identity of a group member by the group signature. At the same time, in the process of signature generation, the group manager cannot know who signed the message unless it performs Open operation to find the signer by traversing.

4.4. Collusion Attack Prevention. Our scheme can resist two kinds of collusion attack. First, multiple group members disclose the key of other members. Second, the group manager divulges the group member's key. For the first case, due to the difficulty of discrete logarithms, even if other group members can discover who generated the group signature, it is impossible to get any information about the key of the signer. For the second case, because the user's secret key is generated by the distributed key generation algorithm, all authorities do not know the user's specific key. In this paper, we need at least T authorities to recover the user's secret key, so the scheme can resist the collusion attack of T authorities in key protection.

5. Application to PDP

This section takes the PDP scheme as an example to apply the multimanager group signature scheme. PDP scheme is divided into two phases: data upload and verification. A PDP scheme is a collection of four algorithms (KeyGen, TagBlock, GenProof, and CheckProof). Among them, the function of KeyGen is to generate the public key and private key of the data owner, and the function of TagBlock is to preprocess the data to be stored. They all exist in the data upload phase. Naturally, GenProof and CheckProof exist in the verification phase. Their function is to generate the proof and verify the proof. Next, we integrate our scheme with PDP and blockchain.

With the rapid development of internet media, a large number of original contents such as text, pictures, audio, and video have been produced, and a large number of copyright certificates are needed. Especially for enterprise users, cloud storage is an effective way to protect digital rights. In this case, cloud service providers or third-party depositors will provide PDP certification. However, in some specific cases, the data owner often lacks the original identification of data copyright, and it is very likely to upload some secondary processed infringing works to cloud storage. If cloud service providers or third-party depositors provide PDP certification for these data, it will become fixed infringement

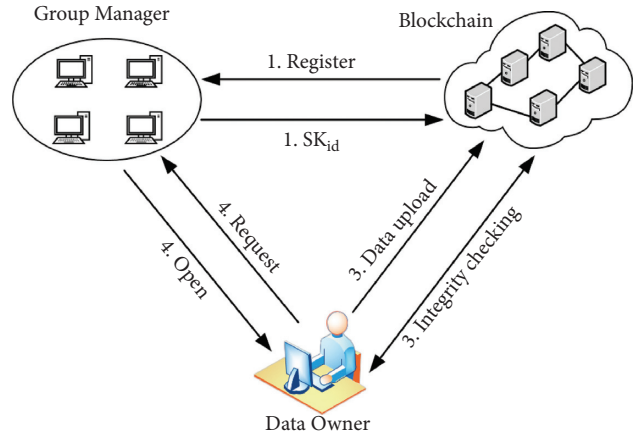

Figure 2: The application of our scheme.

evidence for data owners. Based on the business relationship between cloud service providers and enterprise users, cloud service providers may not want the data owners to know that it is the infringement proof provided by themselves but hope that their privacy will be protected, not that the cloud service providers intend to provide infringement certificates. In this way, it is beneficial for cloud service providers to maintain trust relationship with enterprise users. In this scenario, PDP services can be provided by a third-party depository institution integrating multiple cloud servers, and the cloud storage of multiple service providers can also form a blockchain to solve the privacy problem instead of a single service provider which provides PDP certification. Multicloud service providers provide PDP certification to data owners through the group signature, and key distribution depends on the group manager. It effectively avoids the privacy problem when a single organization provides PDP certification to the data owner. The PDP model based on multiauthorities, ID, and blockchain is shown in Figure 2.

(1) Register: multicloud service providers form the blockchain alliance chain, register with the group manager, and apply for the key.

(2) Data upload: the data owner uploads data to the multicloud alliance chain, which runs the storage algorithm of the PDP scheme and saves the data on cloud storage.

(3) Integrity verification: the data owner or other third-party organizations run the challenge algorithm of the PDP scheme to the cloud server, and the cloud server runs the PDP certification algorithm and returns the proof. The data owner or a third party runs a validation algorithm to verify its integrity. If the verification is successful, the data owner does not know which cloud service provider has completed the signature, so the scheme can protect the privacy of the cloud service provider. If the verification is not successful, go to Step 4.

(4) Open: if the data owner or a third party finds that the data integrity verification fails, the data owner sends a request to the group manager, who performs the

Open operation to determine which cloud service provider implements the signature.

6. Conclusion

In this paper, we propose a multimanager group signature scheme and analyze its security. At the same time, we apply the proposed scheme to the multicloud storage environment based on the blockchain to support the authentication of provable data possession.

Data Availability

No data were used during this study.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (no. 61702067) and Natural Science Foundation of Chongqing (no. cstc2020jcyj-msxmX0343).

References

[1] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system," 2018, https://bitcoin.org/bitcoin.pdf.

[2] M. Swan, "Blockchain thinking: the brain as a decentralized autonomous corporation [commentary]," *IEEE Technology and Society Magazine*, vol. 34, no. 4, pp. 41–52, 2015.

[3] D. L. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Communications of the ACM*, vol. 24, no. 2, pp. 84–90, 1981.

[4] J. Bonneau, A. Narayanan, A. Miller, J. Clark, and E. W. Felten, "Mixcoin: anonymity for bitcoin with accountable mixes," in *Proceedings of the International Conference on Financial Cryptography and Data Security*, pp. 486–504, Christ Church, Barbados, March 2014.

[5] G. Maxwell, "Coinjoin: bitcoin privacy for the real world," 2021, https://en.bitcoin.it/wiki/CoinJoin.

[6] T. Ruffing, P. Moreno-Sanchez, and A. Kate, "CoinShuffle: practical decentralized coin mixing for bitcoin," in *Proceedings of the 19th European Symposium on Research in Computer Security*, pp. 345–364, Wroclaw, Poland, September 2014.

[7] T. Ruffing, P. Moreno-Sanchez, and A. Kate, "P2P mixing and unlinkable bitcoin transactions," in *Proceedings of the Network and Distributed System Security Symposium*, pp. 824–838, NDSS, San Diego, CA, USA, March 2017.

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

[9] E. Ben-Sasson, A. Chiesa, C. Garman et al., "Zerocash: decentralized anonymous payments from bitcoin," in *Proceedings of the 2014 IEEE Symposium on Security and Privacy*, pp. 459–474, IEEE, Washington, DC, USA, May 2014.

[10] N. Shen and M. Adam, "Ring confidential transactions," *Ledge*, vol. 1, no. 1, pp. 1–18, 2016.

[11] T. Ruffing and P. Moreno-Sanchez, "ValueShuffle: mixing confidential transactions for comprehensive transaction privacy in bitcoin," in *Proceedings of the International Conference on Financial Cryptography and Data Security*, pp. 133–154, Sliema, Malta, April 2017.

[12] D. Chaum and E. Van Heyst, "Group signatures," in *Advances in Cryptology*, pp. 257–265, Springer, Berlin, Germany, 1991.

[13] J. Guo, J. Baugh, and S. Wang, "A group signature based secure and privacy-preserving vehicular communication framework," in *Proceedings of the 2007 Mobile Networking for Vehicular Environments*, pp. 103–108, Anchorage, AK, USA, May 2007.

[14] M.-H. Park, G.-P. Gwon, S.-W. Seo, and H.-Y. Jeong, "RSU-Based distributed key management (RDKM) for secure vehicular multicast communications," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 3, pp. 644–658, 2011.

[15] Y. Sun, Z. Feng, Q. Hu, and J. Su, "An efficient distributed key management scheme for group-signature based anonymous authentication in VANET," *Security and Communication Networks*, vol. 5, no. 1, pp. 79–86, 2012.

[16] S. H. Islam, M. S. Obaidat, P. Vijayakumar, E. Abdulhay, F. Li, and M. K. C. Reddy, "A robust and efficient password-based conditional privacy preserving authentication and group-key agreement protocol for VANETs," *Future Generation Computer Systems*, vol. 84, pp. 216–227, 2018.

[17] J. Cui, X. Tao, J. Zhang, Y. Xu, and H. Zhong, "HCPA-GKA: a hash function-based conditional privacy-preserving authentication and group-key agreement scheme for VANETs," *Vehicular Communications*, vol. 14, pp. 15–25, 2018.

[18] A. Shamir, "Identity based cryptosystems and signature schemes," in *Advances in Cryptology*, pp. 47–53, Springer, Berlin, Germany, 1984.

[19] J. Chang, H. Wang, F. Wang, A. Zhang, and Y. Ji, "RKA security for identity-based signature scheme," *IEEE Access*, vol. 8, pp. 17833–17841, 2020.

[20] X. Cheng, S. Zhou, L. Guo, J. Yu, and H. Ma, "An iD-based short group signature scheme," *Journal of Qingdao University (Natural SciEnce Edition)*, vol. 8, no. 3, pp. 554–559, 2012.

[21] Z. Zhang and Y. Ye, "A new id-based threshold group signature scheme," in *Proceedings of the 8th International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 1–4, IEEE, Shanghai, China, September 2012.

[22] L. Ma, "A generic construction of identity-based group signature," in *Proceedings of the 2013 Fourth International Conference on Emerging Intelligent Data and Web Technologies*, pp. 624–626, IEEE, Xi'an, China, September 2013.

[23] S. Pulagara and P. Alphonse, "An intelligent and robust conditional privacy preserving authentication and group-key management scheme for vehicular ad hoc networks using elliptic curve cryptosystem," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 3, pp. 1–10, 2019.

[24] G. Liang, S. R. Weller, F. Luo, J. Zhao, and Z. Y. Dong, "Distributed blockchain-based data protection framework for modern power systems against cyber attacks," *IEEE Transactions on Smart Grid*, vol. 10, no. 3, pp. 3162–3173, 2019.

[25] W. Tong, X. Dong, Y. Shen, and X. Jiang, "A hierrchical sharding protocol for multi-domain IoT blockchains," in *Proceedings of the IEEE International Conference on Communications (ICC 2019)*, pp. 1–6, IEEE, Shanghai, China, May 2019.

[26] Y. Yu, Y. Zhao, Y. Li, X. Du, L. Wang, and M. Guizani, "Blockchain-based anonymous authentication with selective revocation for smart industrial applications," *IEEE*

Transactions on Industrial Informatics, vol. 16, no. 5, pp. 3290–3300, 2020.

[27] F. Tang, S. Ma, Y. Xiang, and C. Lin, "An efficient authentication scheme for blockchain-based electronic health records," *IEEE Access*, vol. 7, pp. 41678–41689, 2019.

[28] G. Ateniese, R. Burns, R. Curtmola et al., "Provable data possession at untrusted stores," in *Proceedings of the CCS'07 14th ACM Conference on Computer and Communications Security 2007*, pp. 598–609, Association for Computing Machinery, Alexandria VA, USA, November 2007.

[29] J. Chang, B. Shao, Y. Ji, M. Xu, and R. Xue, "Secure network coding from secure proof of retrievability," *Science China Information Sciences*, vol. 64, no. 12, Article ID 229301, 2021.

[30] Y. Ji, B. Shao, J. Chang, and G. Bian, "Flexible identity-based remote data integrity checking for cloud storage with privacy preserving property," *Cluster Computing*, vol. 24, no. 3, pp. 1–13, 2021.

[31] F. Chen, F. Meng, T. Xiang, H. Dai, J. Li, and J. Qin, "Towards usable cloud storage auditing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 11, pp. 2605–2617, 2020.