Article

A Consortium Blockchain Wallet Scheme Based on Dual-Threshold Key Sharing

Guojia Li and Lin You *

School of Cyberspace, Hangzhou Dianzi University, Hangzhou 310018, China; liguojia@hdu.edu.cn

* Correspondence: mryoulin@gmail.com

Abstract: In recent years, blockchain has triggered an upsurge in the application of decentralized models and has received more and more attention. For convenience and security considerations, in blockchain applications, users usually use wallets to manage digital assets. The most important data stored in the wallet is the user's private key, which is also the only identification of the ownership of the encrypted digital assets. Once the private key is lost or stolen, it will bring irreparable losses. We proposed a consortium blockchain wallet scheme based on dual-threshold key protection secret-sharing. By splitting and storing the user's wallet private key using a secret-sharing method, we can protect our private keys safely and effectively. Our scheme is based on the application scenario of the consortium blockchain. The peers preset by the consortium blockchain store the user's wallet private key shadow shares, reasonably integrate storage resources, and enhance the solution's anti-attack ability by setting double thresholds.

Keywords: blockchain; consortium blockchain; private key; secret-sharing; verifiability

Citation: Li, G.; You, L. A Consortium Blockchain Wallet Scheme Based on Dual-Threshold Key Sharing. *Symmetry* **2021**, *13*, 1444. <https://doi.org/10.3390/sym13081444>

Academic Editor: Aviv Gibali

Received: 6 July 2021

Accepted: 3 August 2021

Published: 6 August 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Blockchain is a distributed point-to-point network [1], which uses cryptographic algorithms, P2P network architecture, consensus plugin, and other methods to ensure reliable data transmission, storage, and access functions. As a decentralized application, the blockchain has the functions of a distributed ledger and database so that the nodes in a decentralized network can establish a credible distributed system without mutual trust.

Currently, the most popular blockchain application is the cryptocurrency system [2]. Unlike traditional legal currency, cryptocurrency does not require a central control agency, allowing users to trade online without relying on centralized financial institutions. All nodes in the blockchain network can participate in verifying transactions. As blockchain technology has received more and more attention, potential security threats have also increased correspondingly [3].

Transactions in the blockchain system are irreversible. Once digital assets are lost or stolen, they cannot be recovered. The storage of digital assets is related to the security of user assets, and the only thing that can prove the ownership of digital assets is the private key. A digital wallet is a collection of a set of private keys and public keys. The private key generates the public key, and the public key generates the public key address. The corresponding private key can only use the digital asset related to the public key address. The private key can realize the successful transfer of digital assets. The public key and public key address are used to receive the transferred digital assets. The management of digital assets is the management of private keys. The blockchain wallet was born to manage and store keys. Its main functions are to manage users' transaction addresses, initiate transfer transactions, and view transaction records. Each user has a wallet containing multiple secret keys. Therefore, to steal users' digital assets, attackers will attack digital wallets. In recent years, the theft of digital wallets has occurred frequently [4], which has aroused widespread concern. This security threat comes from the private key stored in the digital

wallet. If the attacker stole the user's private key, he can use the digital wallet and transfer all digital assets owned by the wallet.

Digital wallets are very important to the security of the blockchain system, and various wallet schemes have been proposed one after another. The Simplified Payment Verification (SPV) wallet [5] can perform transaction verification without downloading the blockchain, but it is not certain whether the transaction enters the main blockchain. Gutoski [6] proposed a hierarchical deterministic (HD) wallet, which can effectively manage multiple private keys. However, its hierarchical feature makes the private keys have a fixed relationship, leading to some security risks. Rezaeighaleh [7] proposed a new digital scheme to securely back up a hardware wallet relying on the side-channel human visual verification enabled by display screen on a hardware wallet. Wei [8] proposed a new digital signature algorithm and use it to design an online wallet, which can help the user derive the signature without obtaining the user's private key. He [9] proposed a novel cryptocurrency wallet management scheme based on Decentralized Multi-Constrained Derangement (DMCD) to store the keys securely and stably in a decentralized network. Thota [10] presented a secure software wallet which resides on the user's mobile device and using which one participates in a Hyperledger Fabric blockchain network. Boneh and Goldfeder et al. [11] proposed a threshold signature wallet scheme. The private key is divided into n fragments and stored by n participants. The completion of a transaction requires the number of participants greater than the threshold t to sign together, and the Dikshit [12] proposed a scheme based on the identity of the participant gives the participant different weights. Zhou Jian [13] proposed a blockchain wallet protection scheme against single-point failure based on threshold elliptic curve digital signature without a trusted center. This kind of scheme can enhance the security and reliability of the account to a certain extent through the collaborative participation of multiple people to complete a transaction, but how to manage the keys of each participant has also become a difficult problem. In a multi-signature transaction [14–16], M of the N members of the multi-signature transaction address needs to sign to complete the transaction. If someone wants to change the multi-signature transaction strategy, they need to generate a new multi-signature transaction address and script.

In the various wallet schemes mentioned above, most of them have a common problem. Once the device that stores the private key or the tool that assists in recovering the private key is lost, the user cannot recover the wallet anymore. Furthermore, if an attacker can discover the user's wallet key by some mechanism such as brute force, side-channel attack, weak encryption, replay attack or others, then the attacker is able to steal everything from the user's account [17]. Therefore, we propose a consortium blockchain wallet scheme based on dual-threshold key sharing. In our scheme, users can back up their wallets by keeping the private key fragmented by the consortium blockchain peers. The reason we choose to use the consortium blockchain in our scheme is that it has stable preset peers who are efficient to manage and can respond quickly, while the common public chain systems, such as Bitcoin, have a large number of nodes, and it will lead excessive communication overhead between the nodes and other meaningless consumption if all user nodes participate in secret-sharing. The secret-sharing scheme requires some stable and reliable participants. Therefore, the consortium blockchain is suitable for the secret-sharing scheme.

In the following, Section 2 introduces the background of consortium chain and secret-sharing, Section 3 proposes a consortium blockchain wallet scheme based on dual-threshold key sharing, Section 4 carries out security analysis of our scheme and compares it with several similar schemes.

2. Related Information

2.1. Consortium Blockchain

Consortium blockchain [18] is a kind of blockchain that multiple institutions manage. The main user groups of the consortium blockchain are banks, insurance, securities, business associations, and group companies. When the blockchain was born, these com-

panies have generally completed IT and Internet. They realized the blockchain would be very helpful to improve the efficiency of the notarization, settlement, clearing business, and value exchange network in the industrial chain of their circle. However, when trying to use the existing blockchain technology, they found that the processing performance, privacy protection, and compliance of the blockchain could not meet their business needs. On the other hand, if these companies fully adopt Bitcoin's public chain design concept, they will subvert their existing business models and inherent interests and bear great risks. Therefore, they begin to transform the blockchain system that suits them. Consortium blockchain was born. The form of the consortium blockchain is mostly distributed ledgers. Furthermore, the distributed ledgers and consensus of the blockchain solve the main problem: the trust problem of multiple participants in the consortium.

Multiple pre-selected nodes are designated as peers within the consortium blockchain. All pre-selected peers determine the generation of each block. Each organization or institution manages one or more peers, and its data can only be read, written, and sent by different institutions in the system. Each node of the consortium blockchain can only join and exit the network after being authorized by CA (certificate authority). Moreover, user nodes can only participate in the transaction but not the committer process. Any third parties can perform limited queries through the open API of the consortium blockchain [19].

Proof of Work (PoW) [20], Proof of Authorized Share Algorithm (DPoS) [21], Practical Byzantine Fault Tolerant Algorithm (PBFT) [22], are all commonly used consensus algorithms. PoW requires competitive resource consumption, DPoS requires shares proof, while the PBFT needs neither. And PBFT can provide the fault tolerance that does not exceed $1/3$ of the total number of the nodes and can be well adapted to the secret-sharing algorithm to set the threshold. Thus, our scheme uses the PBFT consensus algorithm.

PBFT is an algorithm based on state machine copy replication that aims to solve the problem of ensuring the consistency and correctness of the final decision even when malicious nodes exist in the entire system. Each state machine copy saves the service state and realizes the legal request of customers. In addition to transactions, it can also complete other types of operations and has a wide range of applications.

The PBFT process is shown in Figure 1:

- Request: Client sends a request to the master node 0;
- Pre-Prepare: Node 0 broadcasts to node 1, node 2, and node 3 after receiving Client's request;
- Prepare: Each node records and broadcasts it again after receiving it. Node 3 cannot broadcast due to downtime;
- Commit: In the Prepare phase, if a node receives the same request exceeding $2F$ (F is the number of Byzantine nodes that can be tolerated), it will enter the Commit phase and broadcast the Commit request;
- Reply: In the Commit phase, if one of the nodes receives more than a certain number ($2F + 1$) of the same request, it will give feedback to Client.

Therefore, a consensus can be reached when $N \geq 3F + 1$, where N is the total number of nodes.

Figure 1. PBFT process diagram.

2.2. Shamir's Threshold Secret-Sharing Scheme

In Shamir secret-sharing scheme [23], there are n shareholders $U_i = \{U_1, U_2, \dots, U_n\}$ and a mutually trusted dealer D . To share the secret S into n shares, the dealer D generates a $(t - 1)$ degree polynomial $f(x) \in Z_p$, where P is a prime number. The shared secret is $S = f(0)$, and the dealer computes the secret-sharing shares as $y_i = f(x_i)$ for $x_i \neq 0$, then send the pair (x_i, y_i) to the shareholder U_i . When reconstructing the secret, at least t shares (x_i, y_i) are needed to recover the polynomial $f'(x)$, thus each shareholder can obtain the secret $S = f'(0)$. The scheme consists of two algorithms: share generation and secret reconstruction:

2.2.1. Share Generation

The $(t - 1)$ degree polynomial is defined as $f(x) = a_0 + a_1x^1 + a_2x^2 + \dots + a_{t-1}x^{t-1} \pmod{p}$, and $a_i \in Z_p$, for $0 \leq i \leq t - 1$, $a_{t-1} \neq 0$, the secret $S = f(0) = a_0$.

In a (t, n) secret-sharing scheme, n points need randomly selected as $x_i : 1 \leq i \leq n$, and $x_i \neq 0 \in Z_p$, dealer computes $y_i = f(x_i)$ and sends $s_i = (x_i, y_i)$ to shareholders U_i .

2.2.2. Secret Reconstruction

Suppose that there are t shareholders $U_1, U_2, \dots, U_t$ team up for secret reconstruction. Each shareholder U_i provide the share s_i to the other shareholders. After that, one shareholder has m shares $s_1, s_2, \dots, s_m$ and he can use Lagrange interpolation polynomial to recover $f'(x)$ as:

$$f'(x) = \sum_{i=1}^t s_i \prod_{j=1, j \neq i}^t \frac{x_j - x}{x_j - x_i} \pmod{p}$$

And the secret S can be computed as:

$$f'(0) = \sum_{i=1}^t s_i \prod_{j=1, j \neq i}^t \frac{x_j}{x_j - x_i} \pmod{p}$$

2.3. Harn-Hsu TCSS Scheme

In Harn-Hsu TCSS scheme [24], there are n shareholders $U = \{U_1, U_2, \dots, U_n\}$ and a mutually trusted dealer D . The initial threshold is t and it can be increased to the exact number of shareholders who participate in secret reconstruction. This scheme consists of two algorithms: share generation and secret reconstruction.

2.3.1. Share Generation

The dealer D picks a prime number p and a random symmetric polynomial $F(x, y)$ with degree $t - 1$ as:

$$\begin{aligned} F(x, y) = & a_{0,0} + a_{1,0}x + a_{0,1}y + a_{2,0}x^2 + a_{1,1}xy + a_{0,2}y^2 \\ & + \dots + a_{t-1,0}x^{t-1} + a_{t-2,1}x^{t-2}y + \dots \\ & + a_{0,t-1}y^{t-1} \pmod{p} \end{aligned}$$

where the coefficient $a_{i,j} \in Z_p$, $a_{i,j} = a_{j,i}$, and $\forall i, j \in [0, t - 1]$. The secret $S \in Z_p$ satisfies $s = F(0, 0) + bF(1, 1)$, where $b \in Z_p$. The dealer D picks n different positive integers $x_1, x_2, \dots, x_n$ from Z_p ($x_i \neq 1$) and computes $s_i = F(x_i, y)$, for $i = 1, 2, \dots, n$. Then dealer D distributes each share $s_i(y)$ to the shareholder U_i securely.

2.3.2. Secret Reconstruction

Suppose that m ($t \leq m \leq 1 + \frac{t(t+1)}{2}$) shareholders, for example, $U_1, U_2, \dots, U_m$ want to recover the secret. Each shareholder U_i accesses the public information b and uses its share $s_i(y)$ to compute:

$$w_i = s_i(0) \prod_{j=1, j \neq i}^m \frac{x_j}{x_j - x_i} + bs_i(1) \prod_{j=1, j \neq i}^m \frac{x_j - 1}{x_j - x_i} \mod p$$

Each shareholder U_i sends w_i to the other shareholders. After that, every shareholder has $w_1, w_2, \dots, w_m$ and the secret can be evaluated as:

$$s = \sum_{i=1}^m w_i \mod p$$

However, the paper [25] uses a linear subspace method to successfully attack the TCSS scheme. It is proved that when the threshold is raised from t to m , the participants of $t + 1$ are also enough to reconstruct the secret. Therefore, the TCSS scheme does not have the threshold changeable property, and the increase of threshold is vulnerable to illegal participants.

3. Proposed Scheme

In this section, we propose a consortium blockchain wallet scheme based on dual-threshold key sharing. In our scheme, the private key of the user's wallet is shared and stored by the consortium blockchain peers. Moreover, we use user's biometric key [26] to encrypt the shares in the secret-sharing process. With the particularity of the user's biometric key, only users can perform secret recovery. To deal with potential attack threats, we set a dual-threshold mechanism, it means that a bivariate polynomial is applied for sharing one secret information, and the sharing scheme has two thresholds, t and v ($t = 2v$) which correspond to two sets of sub-secrets, respectively. The two sets of sub-secrets are independent when recovering the secret. In normal operation, the system uses the (t, n) threshold scheme, when the system is attacked or the network fails and cause the number of peers that can honestly respond to users' requests is less than the threshold t , then the (v, n) threshold scheme is enabled. In that case, users can still recover their wallets even when the attacker breaks through several peers.

There is an access mechanism in the consortium chain network, which requires the support of a CA, and all members of the consortium keep a certificate issued by the CA. Peers and users joining the consortium blockchain must be registered and obtain a certificate issued by the CA before they can operate in the consortium blockchain.

In the scheme, there are a user U and peers $P = \{P_1, P_2, \dots, P_n\}$, the threshold t ($t = \frac{2}{3}n$) and v ($v = \frac{1}{3}n$), the threshold $t = 2v$. The user U is both a secret sharer and secret combiner. Peers who exercise committer power are shareholders.

In the secret share phase, we use an asymmetric bivariate polynomial to achieve the dual-threshold secret-sharing. The user divides the secret into sub-secrets and encrypts them as shadow shares, then sends them to peers. Each peer only holds one shadow share, and it can be verified. In the secret reconstruction phase, the user U asks for t or more shadow shares from peers, and each shadow share can be verified for correctness, confirm that all the shadow shares are true and effective. U reconstructs the secret and finally reconstructs the original private key. However, some peers may fail to respond due to failures, maintenance, etc., or some peers may have been attacked and bribed so that they sent the wrong shares. The user U can only obtain m ($t/2 < m < t$) shares, it cannot reach the threshold t to complete the wallet recovery. At this time, the (v, n) threshold scheme in the dual-threshold scheme can be activated.

The biometric key SK [12] held by the user is the key information when recovering the private key. When the user needs to use SK , it can be achieved by extracting personal

biometrics without memory and backup. Additionally, only using the SK can the private key s be recovered correctly in the secret recovery phase.

The notations used throughout the presentation are summarized in Table 1.

Table 1. Notations used in the proposed scheme.

Notation	Description
t	The value of the first threshold
v	The value of the second threshold
n	The number of peers
$H(\cdot)$	SHA-256 hash function
$a \parallel b$	Encoding a and b as strings for a concatenation operation
$a \oplus b$	Encoding a and b as a binary bit string for an XOR operation
pk	The wallet private key
SK	The biometric key of user
sx_i	The sub-secret of (t, n) secret-sharing
sy_i	The sub-secret of (v, n) secret-sharing
x_i	The random number corresponding to sx_i
y_i	The random number corresponding to sy_i
SX_i	The shadow share of sx_i
SY_i	The shadow share of sy_i
X_i	The shadow number of x_i
Y_i	The shadow number of y_i
V_{x_i}	The verification message of (t, n) ss
V_{y_i}	The verification message of (v, n) ss

3.1. Algorithms

The proposed scheme consists of three phases: secret-sharing, peers verification phase and secret reconstruction.

3.1.1. Secret-Sharing Phase

The user U chooses a prime number p and selects an asymmetric bivariate polynomial $F(x, y)$:

$$\begin{aligned}
 F(x, y) &= F_1(x, y) + F_2(x, y) \\
 F_1(x, y) &= a_{0,0} + a_{1,0}x + a_{0,1}y \\
 &\quad + a_{2,0}x^2 + a_{1,1}xy + a_{0,2}y^2 + \dots \\
 &\quad + a_{v-1,0}x^{v-1} + \dots + a_{0,v-1}y^{v-1} \bmod p \\
 F_2(x, y) &= a_{v,0}x^v + \dots + a_{1,v-1}xy^{v-1} \\
 &\quad + a_{v+1,0}x^{v+1} + a_{v,1}x^{v+1}y + \dots + a_{2,v-1}x^2y^{v-1} \\
 &\quad + \dots + a_{t-2,0}x^{t-2} + \dots + a_{t-v-1,v-1}x^{t-v-1}y^{v-1} \\
 &\quad + a_{t-1,0}x^{t-1} + \dots + a_{t-v,v-1}x^{t-v}y^{v-1} \bmod p
 \end{aligned}$$

where $\partial F(x, y) = t - 1$, $a_{i,j} \in Z_p$ and $\forall i, j \in [0, v - 1]$. The secret $pk \in Z_p$, satisfy $pk = F(0, 0)$. (Note: the (t, v, n) threshold requires that the order of x does not exceed $t - 1$, and the order of y does not exceed $v - 1$.)

Then, user chooses positive integers $x_i \in Z_p, y_i \in Z_p, (x_i \neq 0, y_i \neq 0)$, computes the sub-secrets $sx_i = F(x_i, 0), sy_i = F(0, y_i), i = 1, 2, \dots, n$, then uses SK to compute the shares for peer P_i :

- $SX_i = sx_i \oplus SK$.
- $X_i = x_i \oplus H(SK) \oplus H(sx_i)$.
- $SY_i = sy_i \oplus H(SK)$.
- $Y_i = y_i \oplus H(SK) \oplus H(sy_i)$.
- User computes the verification message $V_{x_i} = H(sx_i \parallel x_i), V_{y_i} = H(sy_i \parallel y_i)$.

- Let $m_i = H(SX_i \parallel X_i \parallel V_{x_i} \parallel SY_i \parallel Y_i \parallel V_{y_i})$ be the plaintext, then use the improved the El Gamal signature to sign m_i . Select a large prime number p and set g to be the generator of the group $GF(p)$. Additionally, user selects a random number $l \in [1, p-1]$, $\gcd(l, p-1) = 1$, l is the private key, and computes the modulo inverse of l as $d = l^{-1} \bmod p$. Then user computes $y = g^l \bmod p$, sets (y, g, p) as the public key.
 - The user signs the plaintext, selects a random number $k_i \in Z_p$, then computes $r_i = g^{k_i} \bmod p$.
 - U computes $s_i = (m_i - k_i r_i) d \bmod (p-1)$, the signature of m_i is (r_i, s_i) .
- U sends the secret share $(SX_i, X_i, V_{x_i}, SY_i, Y_i, V_{y_i}, r_i, s_i)$ to the peer P_i . The secret-sharing phase shown in Figures 2 and 3.

Figure 2. Shadow share generation.

3.1.2. Peers Verification Phase

The peer P_i receives the share $(SX_i, X_i, V_{x_i}, SY_i, Y_i, V_{y_i}, r_i, s_i)$ sent by the user, and verifies it:

- P_i computes $m'_i = H(SX_i \parallel X_i \parallel V_{x_i} \parallel SY_i \parallel Y_i \parallel V_{y_i})$;
- if $y^{s_i} r_i^{r_i} = g^{m'_i} \bmod p$ is true, the signature is valid, and the secret share is correct.

Then peer P_i stores (SX_i, X_i, V_{x_i}) and (SY_i, Y_i, V_{y_i}) so that the user can obtain it to recover the secret.

Figure 3. Shadow share distribution.

3.1.3. Secret Reconstruction Phase

When U initiates a wallet private key recovery application to the peers, he should provide the certificate issued by the CA. After confirming the digital identity of the user U, each peer sends the share (SX_i, X_i, V_{x_i}) to the user as Figure 4 shows. If the user receives t or more shares, U can start to reconstruct the secret.

Figure 4. (t, n) secret reconstruction.

U enters the biometric key SK to compute the sub-secret:

- $sx_i = SX_i \oplus SK$.
- $x_i = X_i \oplus H(SK) \oplus H(sx_i)$.
- $V'_{x_i} = H(sx_i \parallel x_i)$.
- if V'_{x_i} is equal to V_{x_i} , the P_i is an honest peer, and the share P_i provided is correct and valid, otherwise it can be judged as malicious peer. Malicious peers will be punished.

After t or more shares are verified correctly, the user obtains t pairs of sub-secrets (sx_i, x_i) , then uses Lagrangian interpolation to reconstruct the secret:

$$\begin{aligned} pk' &= \sum_{i=1}^t sx_i \prod_{j=1, j \neq i}^t \frac{x_j}{x_j - x_i} \bmod p \\ &= \sum_{i=1}^t F(x_i, 0) \prod_{j=1, j \neq i}^t \frac{x_j}{x_j - x_i} \bmod p \\ &= pk \end{aligned}$$

If some peers cannot respond due to failure, maintenance, etc., or some peers may have been attacked and bribed so that they may send the wrong shares. The user U can only obtain m ($t/2 < m < t$) shadow shares, it cannot reach the threshold t to complete the wallet recovery. At this time, the (v, n) threshold scheme in the dual-threshold scheme can be activated. Figure 5 shows the second threshold reconstruction. The user initiates a secondary wallet private key recovery application to m peers that have passed verification, and the peer sends the user (SY_i, Y_i, V_{y_i}) to compute the sub-secret:

- $sy_i = SY_i \oplus H(SK)$.
- $y_i = Y_i \oplus H(SK) \oplus H(sy_i)$.
- $V_{y_i} = H(sy_i \parallel y_i)$.

After that, U obtains t pairs of sub-secrets (sy_i, y_i) and uses Lagrangian interpolation to reconstruct the secret:

$$\begin{aligned} pk' &= \sum_{i=1}^v sy_i \prod_{j=1, j \neq i}^v \frac{y_j}{y_j - y_i} \bmod p \\ &= \sum_{i=1}^v F(0, y_i) \prod_{j=1, j \neq i}^v \frac{y_j}{y_j - y_i} \bmod p \\ &= pk \end{aligned}$$

Figure 5. (v, n) secret reconstruction.

3.1.4. Peers Addition and Deletion

Blockchain nodes can be added or deleted. In response to such situations, we propose the following countermeasures:

- **Join and Exit of User Nodes:**
Whether user nodes join or quit the consortium chain network does not affect the secret share and recovery of the user's wallet private key in the scheme. A new user joins the consortium chain network should perform the secret-sharing scheme, and the user's private key would be stored by the peers, and can be reconstructed normally. When the user logs off the account, the share is also deleted.
- **Join of Peers:**
When a new peer joins the consortium chain network will not affect the secret-sharing that has been performed, but the value of (t, v, n) involved in the subsequent secret-sharing will change accordingly, and the user needs to perform a new threshold secret-sharing after completing a secret reconstruction phase. All the original shadow shares stored in each peer should be destroyed, and peers keep the brand-new shadow shares of the new (t', v', n) threshold secret sharing.
- **Exit of Peers:**
When an original peer exits the consortium blockchain network, there is a possibility that the number of remaining peers is less than the threshold t . At this time, the user can use the second threshold scheme (v, n) to reconstruct the wallet secret key. And after the user finishing the secret recovery, the old shares should be updated. Thus, all the original shadow shares stored in each peer should be destroyed, and peers keep the brand-new shadow shares of the new (t', v', n) threshold secret-sharing.

4. Security Analysis and Scheme Comparison

4.1. Security Analysis

In this section, we will analyze the security to prove the robustness of the proposed scheme against some threats. We assume that the users in the scheme are trusted users.

Theorem 1. *Any subset of participants with t members cannot recover the secret.*

Proof. In the (t, n) threshold scheme, at least t set of shares are required to reconstruct the secret. In the proposed scheme, each committer peer P_i only holds the shadow share (SP_i, SQ_i, X_i) that are generated using $SP_i = sp_i \oplus SK$, $SQ_i = sq_i \oplus H(SK) \oplus H(sp_i)$, $X_i = x_i \oplus H(SK) \oplus H(sq_i)$, and SP_i, SQ_i, X_i need SK to retrieve sp_i, sq_i, x_i respectively. SK is user's biometric extraction key. Thus, sp_i, sq_i, x_i cannot be retrieved by any participant. Therefore, t participants will fail to reconstruct the secret by exchanging information. $\square$

Theorem 2. *Any adversary cannot recover the secret by performing 'Man-in-the-middle attack'.*

Proof. The attacker may intercept the message when the user initiates a secret recovery request to committer peers, and simulates the user to apply for the secret-sharing share to each peer to obtain (SP_i, SQ_i, X_i) . However, the attacker does not have the key SK , and cannot generate the real share sp_i, sq_i, x_i from (SP_i, SQ_i, X_i) to complete the secret recovery which means (SP_i, SQ_i, X_i) has no meaning to the attacker. $\square$

Theorem 3. *Any attacker cannot recover the secret by bribing the committer peer.*

Proof. Even if the attacker bribes peers and obtains the (SP_i, SQ_i, X_i) kept by committer peer, he cannot compute the sp_i, sq_i, x_i from (SP_i, SQ_i, X_i) to obtain any secret-related information about the secret due to lack of the biometric binding key SK . $\square$

Theorem 4. *Illegal peers can be identified in the proposed scheme.*

Proof. When the user wants to recover the secret, at least t committer peers need to respond and send the shares (SP_i, SQ_i, X_i) , and the user uses SK to compute t pairs of shares sp_i, sq_i, x_i . Then the correct secret can be recovered.

Suppose that when the committer peer sends information, replace SP_i with SP'_i . The user computes:

$sp'_i = SP'_i \oplus SK$ obtains sp'_i , which is not equal to sp_i ;

$sq'_i = SQ_i \oplus H(SK) \oplus H(sp'_i)$ obtains sq'_i , and $sq'_i \neq sq_i$;

$x'_i = X_i \oplus H(SK) \oplus H(sq'_i)$ obtains x'_i , and $x'_i \neq x_i$;

the verification information $V'_i = H(sp'_i \parallel sq'_i \parallel x'_i)$ is not equal to V_i , so the verification fails, and the peer P_i may be judged as an illegal peer.

Suppose the share (SP_i, SQ_i, X_i) that provided by committer peer can pass the user's verification while the share has been changed, which means that the peer P_i has found two different numbers with the same hash value. And that is not achievable in polynomial time. $\square$

Theorem 5. Shareholders in this scheme also have verification capabilities.

Proof. In secret-sharing phase, user U shares the secret, generates share $(SP_i, SQ_i, X_i, V_i, r_i, s_i)$ and sends it to the committer peer P_i . If there is an attacker A intercepts the share during the communication, tampers the share then sends $(SP'_i, SQ'_i, X'_i, V'_i, r'_i, s'_i)$ to the committer peer P_i .

The committer peer P_i receives the share, compute the plaintext content as $m'_i = H(SP'_i \parallel SQ'_i \parallel X'_i \parallel V'_i)$ with the public information (y, g, p) verify the signature, and P_i find out the equation $y^{s'_i} r'^{r'_i}_i = g^{m'_i} \pmod{p}$ is not satisfied. The share is considered fake, P_i refuses to accept the share. $\square$

Theorem 6. Our scheme allows the user to update the biometric extraction key SK .

Proof. If the user wants to replace the key SK with SKN , computes:

$$M_1 = SK \oplus SKN$$

$$M_2 = H(SK) \oplus H(SKN)$$

User sends M_1 and M_2 to every committer peer, each of them performs the following operations:

$$SP'_i = SP_i \oplus M_1 = sp_i \oplus SK \oplus SK \oplus SKN = sp_i \oplus SKN$$

$$SQ'_i = SQ_i \oplus M_2 = sq_i \oplus H(SK) \oplus H(sp_i) \oplus H(SK) \oplus H(SKN) = sq_i \oplus H(SKN) \oplus H(sp_i)$$

$$X'_i = X_i \oplus M_2 = x_i \oplus H(SK) \oplus H(sq_i) \oplus H(SK) \oplus H(SKN) = x_i \oplus H(SKN) \oplus H(sq_i)$$

The committer peer replaces (SP_i, SQ_i, X_i) with (SP'_i, SQ'_i, X'_i) , to achieve the user's update of the biometric key. Here every user can update its key SK without let committer peers knowing about the secret and the committer peer is given relief from storing the secret. $\square$

4.2. Schemes' Comparison

In this section, our scheme is compared with some existing blockchain private key protection schemes that use other technologies, and it is summarized in Table 2.

Gutoski et al. [6] proposed a new hierarchical deterministic wallet, which makes it impossible for an attacker to easily recover the master private key while obtaining the master public key and any sub-private key. However, in this scheme, users still need to store the master private key used to recover all private keys. Once the master private key is lost, the user cannot retrieve the property.

Goldfeder et al. [11] proposed a scheme of threshold signature scheme compatible with Bitcoin's signature using the Elliptic Curve Digital Signature Algorithm. They provided a security policy of shared control of a wallet in which each player obtains only a single share. Dikshit et al. [12] proposed an extend the weighted threshold ECDSA scheme. Zhou

Jian et al. [13] proposed a blockchain wallet protection scheme based on threshold ECDSA without a trusted center. These threshold ECDSA requires participants to interact during the preparation phase of the scheme to determine some of the parameters involved in the scheme. Thus, the threshold ECDSA scheme is suitable for a group to conduct transactions and can solve the problem of loss of private keys in the blockchain, but it is not perfect for protecting the private key of a single user.

Table 2. Schemes' comparison.

Scheme	Collusion Resistance	Single Peer Failure	Recoverability	Anonymity	Single Peer Control
Threshold signature scheme	✓	✓	✓	×	×
Multi-signature scheme	✓	✓	×	×	×
Hierarchical deterministic wallet	✓	×	✓	✓	✓
Proposed scheme	✓	✓	✓	✓	✓

We propose a consortium blockchain wallet scheme based on dual-threshold key sharing. In the traditional VSS scheme, illegal participants may submit false information, which will lead to the generation of false secrets and the inability to recover the correct secret. A cheating detection mechanism can only check whether the recovered secret is correct. If it is wrong, in the next secret reconstruction, some existing participants will be excluded until the correct secret is recovered. The cheater identification mechanism realizes to check the authenticity of the information submitted by participants before reconstructing the secret, avoiding the situation of reconstructing the secret multiple times.

Our scheme is applied to the consortium blockchain and realizes the secret-sharing between users and peers. Users can verify shadow shares submitted by the peers and identify illegal peers. Only when the user registers the consortium blockchain account, a third-party trusted center CA is required. In the secret recovery process, there is no third party participates in, the user behavior is anonymous, and the sharing and recovery of secrets are performed by the user himself. Our scheme adopts a verifiable dual-threshold secret-sharing method, which has the recoverability of secrets and the ability to resist single-point failures, to complete the two-way verification of users and peers, and we use the dual-threshold to deal with the unexpected situation of more than one third of the peers' failures in the PBFT consensus algorithm. And we use the biometric key of the user to achieve secret recovery, without the need to memorize and store the passwords.

5. Conclusions

We propose a consortium blockchain wallet scheme based on dual-threshold key protection secret-sharing to solve the problem of recovering the lost private key of the blockchain. Our scheme achieves the secret-sharing between the users and the peers in the consortium blockchain. It can recover the users' wallet's private key within the consortium blockchain by a verifiable dual-threshold secret-sharing method if users lose their wallet private keys. Furthermore, we have proposed the idea of using biometric key to encrypt the shares, which can make secret recovery more secure. Thus, users can recover their wallets safely. Our proposed scheme can resist conspiracy attacks from the consortium blockchain and man-in-the-middle attacks in our security analysis. It can meet the requirements of verifiable threshold secret-sharing. The shadow share authentication mechanism is used to prevent the internal personnel from deceiving, and the threshold feature can resist external network attacks, so it is a more secure solution. And we have overcome the problem that the wallet cannot be recovered when more than 1/3 of the peers fail through the dual-threshold mechanism. However, how to ensure the secure transmission of shadow sharing between nodes of the blockchain and reduce the computational overhead in our scheme remains to be studied.

We will continue to focus on the combination of threshold cryptography and blockchain technology to solve secret key management issues in our future work. We will explore

how to integrate the threshold secret-sharing with the blockchain wallet system more efficiently and expand the application scenarios from the consortium blockchain for more blockchain applications.

Author Contributions: Conceptualization, G.L.; methodology, G.L.; writing—original draft preparation, G.L.; writing—review and editing, L.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (No. 61772166) and the Key Program of the Natural Science Foundation of Zhejiang province of China (No. LZ17F020002).

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

Abbreviations

The following abbreviations are used in this manuscript:

SPV	Simplified Payment Verification
HD	Hierarchical Deterministic
PoW	Proof of Work
DPoS	Delegated Proof of Stake
PBFT	Practical Byzantine Fault Tolerant Algorithm

References

1. Nofer, M.; Gomber, P.; Hinz, O.; Schiereck, D. Blockchain. *Bus. Inf. Syst. Eng.* **2017**, *59*, 183–187. [[CrossRef](#)]
2. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. *Decentralized Business Review*. 2008. Available online: <https://www.google.com.hk/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKewibtt7khJzyAhXOQfUHHfkzBTgQFnoECAYQAw&url=https%3A%2F%2Fbitcoin.org%2Fbitcoin.pdf&usg=AOvVaw05-4mYD7EyyKjwcHh8i0Vw> (accessed on 3 August 2021).
3. Zhu, L.; Gao, F.; Shen, M.; Li, Y.; Zheng, B.; Mao, H.; Wu, Z. Survey on Privacy Preserving Techniques for Blockchain Technology. *J. Comput. Res. Dev.* **2017**, *54*, 2170–2186.
4. Peterson, B. Thieves stole potentially millions of dollars in bitcoin in a hacking attack on a cryptocurrency company. *Bus. Insid.* **2017**, *6*, 12–13.
5. Dai, W.; Deng, J.; Wang, Q. SBLWT: A secure blockchain lightweight wallet based on trustzone. *IEEE Access* **2018**, *6*, 40638–40648. [[CrossRef](#)]
6. Gutoski, G.; Stebila, D. Hierarchical deterministic bitcoin wallets that tolerate key leakage. In *Proceedings of the International Conference on Financial Cryptography and Data Security*, San Juan, Puerto Rico, 26–30 January; Springer: Berlin/Heidelberg, Germany, 2015; pp. 497–504.
7. Rezaeighaleh, H.; Zou, C.C. New secure approach to backup cryptocurrency wallets. In *Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM)*, Waikoloa, HI, USA, 9–13 December 2019; pp. 1–6.
8. Wei, Q.; Li, S.; Li, W.; Li, H.; Wang, M. Decentralized Hierarchical Authorized Payment with Online Wallet for Blockchain. In *Proceedings of the International Conference on Wireless Algorithms, Systems, and Applications*, Honolulu, HI, USA, 24–26 June 2019; Springer: Cham, Switzerland, 2019; pp. 358–369.
9. He, X.; Lin, J.; Li, K.; Chen, X. A novel cryptocurrency wallet management scheme based on decentralized multi-constrained derangement. *IEEE Access* **2019**, *7*, 185250–185263. [[CrossRef](#)]
10. Thota, A.R.; Upadhyay, P.; Kulkarni, S.; Selvam, P.; Viswanathan, B. Software Wallet Based Secure Participation in Hyperledger Fabric Networks. In *Proceedings of the 2020 International Conference on COMMunication Systems & NETworkS (COMSNETS)*, Bangalore, India, 7–11 January 2020; pp. 1–6.
11. Boneh, D.; Gennaro, R.; Goldfeder, S. Using level-1 homomorphic encryption to improve threshold dsa signatures for bitcoin wallet security. In *Proceedings of the International Conference on Cryptology and Information Security in Latin America*, Havana, Cuba, 20–22 September 2017; Springer: Cham, Switzerland, 2017; pp. 352–377.
12. Dikshit, P.; Singh, K. Efficient weighted threshold ECDSA for securing bitcoin wallet. In *Proceedings of the 2017 ISEA Asia Security and Privacy (ISEASP)*, Surat, India, 29 January–1 February 2017; pp. 1–9.
13. Jian, Z.; Ran, Q.; Liyan, S. Securing blockchain wallets efficiently based on threshold ECDSA scheme without trusted center. In *Proceedings of the 2021 Asia-Pacific Conference on Communications Technology and Computer Science (ACCTCS)*, Shenyang, China, 22–24 January 2021; pp. 47–51.
14. Aitzhan, N.Z.; Svetinovic, D. Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams. *IEEE Trans. Dependable Secur. Comput.* **2016**, *15*, 840–852. [[CrossRef](#)]

15. Maxwell, G.; Poelstra, A.; Seurin, Y.; Wuille, P. Simple schnorr multi-signatures with applications to bitcoin. *Des. Codes Cryptogr.* **2019**, *87*, 2139–2164. [[CrossRef](#)]
16. Han, J.; Song, M.; Eom, H.; Son, Y. An efficient multi-signature wallet in blockchain using bloom filter. In Proceedings of the 36th Annual ACM Symposium on Applied Computing, Korea, 22–26 March 2021; pp. 273–281.
17. Pal, O.; Alam, B.; Thakur, V.; Singh, S. Key management for blockchain technology. *ICT Express* **2019**, *7*, 76–80. [[CrossRef](#)]
18. Jiang, Y.; Ding, S. A high performance consensus algorithm for consortium blockchain. In Proceedings of the 2018 IEEE 4th International Conference on Computer and Communications (ICCC), Chengdu, China, 7–10 December 2018; pp. 2379–2386.
19. Androulaki, E.; Barger, A.; Bortnikov, V.; Cachin, C.; Christidis, K.; De Caro, A.; Enyeart, D.; Ferris, C.; Laventman, G.; Manevich, Y.; et al. Hyperledger fabric: A distributed operating system for permissioned blockchains. In Proceedings of the Thirteenth EuroSys Conference, Porto, Portugal, 23–26 April 2018; pp. 1–15.
20. Gervais, A.; Karame, G.O.; Wüst, K.; Glykantzis, V.; Ritzdorf, H.; Capkun, S. On the security and performance of proof of work blockchains. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016; pp. 3–16.
21. Saad, S.M.S.; Radzi, R.Z.R.M. Comparative Review of the Blockchain Consensus Algorithm Between Proof of Stake (POS) and Delegated Proof of Stake (DPOS). *Int. J. Innov. Comput.* **2020**, *10*. [[CrossRef](#)]
22. Sukhwani, H.; Martínez. Performance modeling of PBFT consensus process for permissioned blockchain network (hyperledger fabric). In Proceedings of the 2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS), Hong Kong, China, 26–29 September 2017; pp. 253–255.
23. Shamir, A. How to share a secret. *Commun. ACM* **1979**, *22*, 612–613. [[CrossRef](#)]
24. Harn, L.; Hsu, C.F. Dynamic threshold secret reconstruction and its application to the threshold cryptography. *Inf. Process. Lett.* **2015**, *115*, 851–857. [[CrossRef](#)]
25. Jamshidpour, S.; Ahmadian, Z. Security analysis of a dynamic threshold secret sharing scheme using linear subspace method. *Inf. Process. Lett.* **2020**, *163*, 105994. [[CrossRef](#)]
26. Zhang, N.; Zang, Y.L.; Tian, J. The integration of biometrics and cryptography—A new solution for secure identity authentication. *J. Cryptologic Res.* **2015**, *2*, 159–176.