Future Generation Computer Systems 154 (2024) 330–343

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

A secure multi-party payment channel on-chain and off-chain supervisable scheme

Ke Xiao *, Jiayang Li, Yunhua He, Xu Wang, Chao Wang

School of Information Science and Technology, North China University of Technology, Beijing, 100144, China

ARTICLE INFO

Keywords:
Blockchain
Supervision
Payment channel
Smart contract
Off-chain transaction
Scalability

ABSTRACT

With the increasing demand for digital currencies and other blockchain transactions, the contradiction between the growing demand and the limited blockchain space has become increasingly prominent, and the scalability of the blockchain urgently needs to be resolved. The existing off-chain payment channel solutions do not give enough consideration to the multi-party parallel transaction process, and the transaction process lacks supervision, making it difficult to meet the actual demand. In this paper, we propose a multi-party secure, flexible, concurrent and supervisable off-chain payment channel scheme. Our scheme not only optimizes the current transaction process so that the participants and channel balance within the channel can be flexibly changed without affecting the normal transaction process, but also designs a multi-level collaborative supervision mechanism from multiple perspectives of positive and negative, active and passive, and multiple layers of on-chain and off-chain to ensure the safety and convenience of the transaction process. The security of the scheme was also analyzed, especially the possible collusion in the channel. We simulated the operation of the channel in the blockchain simulation and testing software Simblock and tested the scheme from various aspects. The experiments show that our scheme is able to achieve secure supervision within the channel with a modest overhead (about 15%). Testing of the mentioned smart contracts further illustrates the feasibility of the scheme.

1. Introduction

The gradual maturation of digital cryptocurrency and blockchain technology has captured the attention of an increasing number of individuals, with large-scale blockchain applications becoming an increasingly pressing demand. One critical area within this process is payments. According to Statista, the global cryptocurrency market is projected to grow at a rate of 14.40% annually (between 2023–2027), culminating in a market size of $64.87 billion in 2027 [1]. However, traditional blockchain technology has inherent limitations, as it is designed to support only a limited number of transactions per block. In order to enable nodes to efficiently verify and confirm transactions in order to reach consensus, block size cannot be indefinitely increased, which constrains transaction growth to some extent. In other words, existing blockchain systems suffer from low scalability and are unable to keep pace with the growing demand for transactions.

Off-chain payment channel technology, as the layer II of distributed ledger, is a panacea for blockchain scalability problems. The conventional off-chain payment channel assumes the form of a logical channel between two users on the blockchain network who wish to transact. Users can transfer pre-deposited amounts within the channel and then post the final balance as a single transaction, without the necessity of recording each individual transaction on the blockchain. This methodology significantly reduces the on-chain space required for the same transactions, thereby unlocking the possibility of small transactions with high frequency and large scale.[2]. Lightning Network(LN) [3] and Raiden Network(RN) [4] are used in Bitcoin and Ethereum, respectively, to establish off-chain payment channels between nodes to achieve fast peer-to-peer transactions. The chain only needs to record the opening and closing transactions of the channel, which greatly expands the volume of transactions that can be accommodated in the same block and improves the operational efficiency of the blockchain. The Bitcoin transaction process ensures security and privacy through various technologies such as Hash Lock and Time Lock, while the Monitoring Service is used in Ethereum to guarantee security with a slight loss of privacy.

Due to the tripartite relationship of blockchain scalability, security and decentralization, enhancing one of them will have a negative impact on the other two. That is, enhancing the scalability of blockchain

* Corresponding author.
E-mail addresses: xiaoke@ncut.edu.cn (K. Xiao), lijiayang@mail.ncut.edu.cn (J. Li), heyunhua@ncut.edu.cn (Y. He), wangxu@ncut.edu.cn (X. Wang), wangchao.andy@gmail.com (C. Wang).

https://doi.org/10.1016/j.future.2024.01.012
Received 3 July 2023; Received in revised form 13 November 2023; Accepted 10 January 2024
Available online 11 January 2024

will inevitably lead to a decrease in the overall security and decentralization characteristics [5,6]. Therefore, in order to improve scalability, which is considered the biggest long-term threat to the viability of blockchain technology, it is necessary to find the right trade-off among the three in order to maintain the stability of the triangle.

1.1. Motivation

Currently, off-chain payment channels are being employed in various domains. Distributed energy trading, in particular, aligns perfectly with the off-chain payment channel technology due to its attributes of high transaction volume, frequent transactions, and small individual transaction amounts. As a result, the off-chain payment channel technology has found extensive application in this field [7]. Energy trading entities can leverage the blockchain's tamper-evident feature to establish transaction intentions on the chain while conducting off-chain payments to finalize the process. Some examples of using off-chain payment channel technology also exist in the Internet of Vehicles for gasoline/charging payments [8], parking payments [9]. However, with the increase in transaction scale, the current dual-user channel can no longer meet the demand volume of increasing transactions.

A payment channel network (PCN) is formed by connecting multiple payment channels together. Users can make cross-channel transactions through users or Payment Channel Hubs (PCHs) with whom they have established channels [10]. This improves the utilization of a single channel to some extent, but it also raises other problems, such as privacy protection [11], routing problems of cross-channel payments [12], rebalancing problems after the funds in the channels are exhausted [13], and the problem of forced stopping of transactions caused by the sudden offline of intermediate channels. Several research have attempted to convert multiparty payments into multiple directly connected two-party transactions using the same intermediate payment channel, avoiding the choice of connection channels [14,15]. Nevertheless, the existing multi-party payment scheme remains fundamentally rooted in traditional bilateral payment channels and fails to tackle the overhead issue that hampers the scalability of the payment process on a larger scale.

The Multi-party Payment Channel(MPC) extends the channel that was originally only available to both parties to provide the possibility for multiple parties to transact in one channel at the same time. Some of the solutions, however, undermine the advantages of off-chain payment channels. Schemes in [16] can contain an almost unlimited number of parties in one channel, but it designed to let the payee node, such as the cafe, be the only central node of MPC, which completely destroys the advantages of decentralization of the blockchain system and the direct transaction process between non-central nodes within the channel is not considered. Meanwhile, the channel maintenance problem brought by infinite nodes also deserves further consideration. In addition, it is crucial to note that the existing payment channel schemes all lack effective supervision. Although fraud prevention contracts are proposed in [17] and the supervisory node is proposed in [18], the supervisory process is not comprehensive, and giving excessive authority to the supervisor without controlling it is itself an act that jeopardizes the security of the channel.

To summarize, current channel solutions face the following challenges:

(1) Insufficient consideration of the multi-party payment channel transaction process and it is difficult to realize off-chain transactions that are flexible and reliable and fully utilize the advantages of the blockchain.

(2) Incomplete supervision of the security of transactions in the channel and lack of collaborative supervision from multiple perspectives and at multiple levels.

1.2. Contribution

In order to solve the above problems, we propose a novel multi-party payment channel scheme, which aims at ensuring security and realizing effective supervision both on and off the chain. This scheme revolutionizes the traditional off-chain transaction process and introduces a series of innovative mechanisms, including dual-supervisory users mechanism, reverse-supervision mechanism, etc. Meanwhile, it combines with smart contracts to realize active-passive multi-level on-chain and off-chain collaborative supervision. Specifically,

- We innovatively optimize the previous multi-party payment channel transaction process by designing a four-stage loop of rounds process in the channel, where channel participants can trade in parallel and freely join or exit in each round. Offline or malicious users will be cleared periodically to avoid additional channel operation cost by channel threshold mechanism.
- We firstly design multi-level supervision mechanisms to achieve comprehensive supervision of transactions and transaction participants. The supervisory users supervise the trading users and each other while trading users reverse-supervise the supervisory users which makes both parties form checks and balances. Supervisory smart contracts are designed on-chain to guarantee the identity security of the channel participants and the reliability of the transaction process.
- We also make a detailed analysis and proof of the security of the scheme, especially for possible collusion issues. We also evaluated the performance of the channel on blockchain simulation software Simblock and the results showed that our scheme is effective.

The remainder of the paper is organized as follows. In Section 2, we review the latest research. We introduce the models and goals of our scheme in Section 3. In Section 4, we present our scheme, a multi-party payment channel scheme. And we analyze and evaluate our scheme in Sections 5 and 6. In Section 7, we discuss our scheme with its limitations and we summarize our paper in Section 8.

2. Related works

In this section, we overview some recent research work on solutions of scalability, multi-party transactions and the scheme of supervision.

2.1. Solutions of scalability

The problem of blockchain scalability is a major key factor limiting the development of blockchain, and many experts and scholars have proposed corresponding solutions such as replication, sidechain, sharding or cryptographic methods to solve this problem.

In [19], a scalable protocol, FileInsurer, deploys dynamic replication (DRep) to support the flexible storage of the files on-chain and off-chain, which reduces the pressure on on-chain processing and solving the scalability problem. And this protocol also designed to resist Sybil attacks, further enhancing security. A scheme for blockchain sidechains for extending blockchain transactions is described in [20], where transactions are always rooted in locked transactions in the main chain. Once the proof of a locked transaction is performed on the sidechain, participants can start using the asset by making a transaction on the sidechain. And the latest state of the sidechain must be proven to unlock the coins on the main chain. Hong et al. [21] proposed a novel off-chain cross-shard mechanism to provide efficient cross-shard database services, delegating a large amount of cross-shard transaction data exchange to several nodes, improving the throughput of blockchain database services and solving the scalability problem of blockchain to some extent. Considering the implementation of transactions without the use of digital signatures, Zhong et al. [22] proposed a secure versatile light payment (SVLP) scheme that effectively enhances

the flexibility of the payment and refund process and utilizes basic cryptographic primitives (one-way function) to secure the transaction process.

Above solutions alleviate the blockchain scalability problem to some extent, but they are more or less characterized by excessive overhead, slow transaction speed, insufficient technical security and other problems compared with the off-chain channel technology, which is less applicable to be applied to off-chain transactions. Therefore, our paper focuses on blockchain off-chain channel technology to solve the scalability problem in the transaction process.

2.2. Multi-party transaction

In order to extend the performance of traditional payment channels, some scholars have made studies for channel payment networks. To solve the routing problem of notification payment channel networks, Zhang et al. [12] propose a distributed robust payment routing protocol RobustPay+ that constructs two node disjoint paths to enhance the robustness of payment routing in PCNs, minimizes the worst-case transaction cost under timeliness and feasibility constraints. Other scholars have solved the same problem of multi-user transactions by establishing multi-party payment channels. Chen et al. [14] propose a payment channel scheme that supports simultaneous transactions of multiple people, converting multiparty payments into multiple directly connected two-party transactions using the same intermediate channel for payment. In [18], Ge et al. introduced conditional and redemption payments within the multi-party channel, changed the channel state update process from serial to parallel, and used a greedy routing algorithm based on graph embedding in the channel network to enhance the efficiency of the transaction process. Lei et al. [16] proposed a multi-node payment channel scheme MPC, in which the recipient acts as the central node within the channel and connects other users within the channel to achieve multi-party transaction transfers, avoiding the overhead of adding duplicate channel information to the blockchain and reducing the burden on the blockchain. In MPC, the need for simultaneous multi-user transactions is met, but there are still issues that compromise the decentralized nature of the blockchain and lack of security.

In addition, some researchers have designed smart contracts for the transaction process to further optimize the efficiency of the transaction. Górski et al. [23] introduced a smart contract design and implementation model based on public declarations of verification rules in a renewable energy exchange scenario, which has good reconfigurability to adapt to different transaction types, and at the same time exploits polymorphism to give the possibility of reusing smart contracts. To address the low sustainability caused by channel balance depletion in channel payment networks, Hong et al. [13] proposed an asynchronous rebalancing-based channel protocol CYCLE to design smart contract-based asynchronous update rules for each channel's global offset, avoiding channel freezing caused by rebalancing protocols. However, current smart contracts do not address the entire process from transaction to transaction, and there is a lack of smart contracts for off-chain payment channels.

2.3. Supervisory mechanisms

Some research scholars maintain the security of the channel by introducing some high-security techniques. Lind et al. [24] proposed a Trusted Execution Environment(TEE)-based off-chain payment channel that uses a TEE-protected trusted vault to maintain the funds deposited in the off-chain payment channel with asynchronous access to the blockchain, enabling dynamic management of the amount in the channel and enhancing the security and flexibility of off-chain transactions. Pan et al. [17] propose a multiplexed payment channel scheme to build an out-of-channel payment system enabling multiple payments in one channel, and by introducing a transaction supervisor in the channel and designing a fraud proof mechanism to constrain the trading users in the channel, the supervision of the transaction process is achieved.

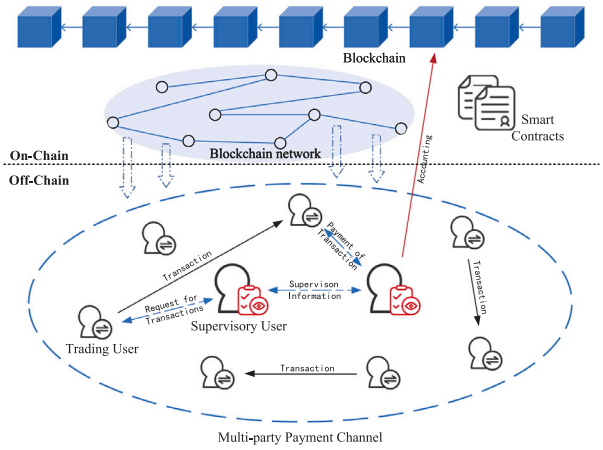

Fig. 1. System Model.

Some studies have also considered introducing a third-party into the channel trading process as the outside supervisor to oversee the proper functioning of the channel. McCorry et al. [25] consider the case where a malicious channel participant seeks to benefit from reversing the collective authorized state in a channel by executing a fork attack, and propose a protocol PISA for responsible third-party participation in transactions, where an online trusted third party monitors the channel state and disputes the improper channel state on behalf of offline users. Du et al. [26] introduce watchtowers as third-party regulators of cross-channel transactions in channel payment networks and demonstrate their reliability through game theory. Ge et al. [27] propose a multi-party channel construction scheme, Magma, which effectively enables secure transactions within the channel by introducing an untrusted third-party operator to maintain the channel and providing a reporting means to restrict operator misbehavior, balancing the robustness and flexibility of the off-chain channel.

In the above two ideas, the introduction of security technology is only for specific segments, and the trustworthiness design of third-party regulators is also questionable. There is a lack of a comprehensive security supervision scheme to guarantee the safe and stable operation of the payment channel under the blockchain.

3. Model construction and design goals

In this section, we define the system model and threat model of the scheme and describe the design goal of the scheme.

3.1. System model

The system model of this scheme is shown in Fig. 1. It consists of two layers of structure, Layer I: on-chain and Layer II: off-chain. There is one entity in Layer I, the blockchain. Nodes in the blockchain network can establish off-chain payment channels by triggering smart contracts, and other contracts on-chain also have the responsibility of maintaining the operation of transactions and implementing supervision. There are two types of entities in Layer II, trading users and supervisory users. Supervisory users are elected from trading users, and their essence are still the trading users. Within the off-chain payment channel, transactions can be conducted between trading users to conduct payment by transferring their account balances of the channel. The actual transaction process consists of the transaction user initiating a transaction request to a supervisory user; the request is validated and synchronized

to another supervisory user for supervision; and finally the supervisory users complete the payment.

We define the user group in the channel during the round i as $N_i = \{node_1, node_2, \ldots, node_n\}$ and each user involved in the transaction as $node_j = \{id_j, d_j, T\}$ where id_j is the identifier of a trading user, d_j stands for users' account balance in the channel, T is channel threshold of the user. The channel state of the ith round is defined as $state_i = \{i, N_i, S_a, S_b, Dep\}$, and the channel itself is defined as $PC = (num, state_i, \Omega)$. S_a and S_b are the supervisory users of the channel, and Dep is the deposit of them. num and Ω are the identifier of the channel and the channel threshold.

The traditional off-chain payment channel is participated by both payment parties, and both transfer the amount to be transacted to the channel account balance when the channel is established, make payments by posting the latest state in the channel, and return the latest balance to the user blockchain address at the end of the channel to complete the off-chain in-channel transactions.

The design of the multi-party off-chain payment channel is different from the traditional two-party channel, but it can also be generally divided into three processes: establishment, transaction, and closure. The multi-party channel scheme proposed in this paper further subdivides the transaction process into four stages: Channel Preparation stage, Transaction stage, Transaction Settlement stage, and Channel Update stage due to the design of the supervisory mechanism.

- *Channel Establishment*. Users with transaction needs set up a payment channel by triggering the associated smart contracts. Each trading user deposits a certain amount in a channel as the account balance of the channel.
- *Transaction*. During the Channel Preparation stage, the main focus is on supervisory users' election. In the Transaction stage, the channel participants (i.e., trading users and supervisory users) make transactions. In the transaction settlement stage, the supervisory users integrate and broadcast the current channel state for the trading users to verify. Trading users choose to join in or exit, and the state of the channel is determined by whether it needs to be closed in the Channel Update stage. The duration of the four stages are Δ_P, Δ_T, Δ_S, Δ_U.
- *Channel Closure*. If the channel does not meet the condition of continued existence, it will enter the process of *Channel Closure*. The user's channel account balance will be returned to the user's blockchain address, the supervisory users' deposit will be returned at the same time, and the channel will be closed.

3.2. Threat model

In our scheme, the blockchain is assumed to be honest, that is it honestly behaves just like the scheme designed. To some extent, it can be regarded as a reliable entity without disrupting transactions or colluding with users. The trading users and supervisory users are assumed to be normal-and-curious, that is on the one hand they honestly participate in channel operations to make sure their transaction requests will be correctly processed and the other users' maliciously behavior will be punished, on the other hand, they try to reach additional incomes by any probably measurement. The number of honest users in the channel is not less than half of the total number of participants in the channel. This scheme does not take into account the offline of users in the channel due to some special circumstances. Users who want to exit the channel will actively leave the channel at the end of each transaction round. The communication process within the channel is assumed reliable, and information sent by the user must be received.

Since none of the participants involved in the channel transaction process is fully trustworthy, the threat model involves two entities, the ordinary trading users and the supervisory users. Therefore, we give the following definitions:

Definition 1. The behavior Bhv_U of the ordinary trading user is

$$Bhv_U = Need_U + Constraint_U + Incentive_U. \tag{1}$$

If each item in the Bhv_U is satisfied, the ordinary trading user performs the behavior as the scheme designed in the channel.

In Definition 1, $Need_U$ stands for the common transaction needs for a trading user, $Constraint_U$ stands for supervisory users' supervision, $Incentive_U$ stands for the rewards from the forfeiture of margin for misconduct of supervisory users.

Definition 2. The behavior Bhv_S of the supervisory user is

$$Bhv_S = Need_S + Constraint_S + Incentive_S. \tag{2}$$

If each item in the Bhv_S is satisfied, the supervisory user performs the behavior as the scheme designed in the channel.

In Definition 2, $Need_S$ stands for the supervision for a supervisory user, $Constraint_S$ stands for reverse supervision from trading users and the mutual supervision from other supervisory user, $Incentive_S$ stands for the rewards from the supervision and the operation of channel.

3.3. Design goal

Under the mentioned system model and threat model, our design goal is to design a secure, efficient and supervised multi-party payment channel scheme. The main objectives are as follows.

- **Secure and efficient payment channel.** The proposed scheme should enable an off-chain payment channel jointly established by multiple transaction participants and enable concurrent and reliable secure payments within the channel. The channel participants should be able to join or exit the channel at will without disturbing the normal operation of the channel, and the user account balance within the channel can be dynamically changed to ensure flexibility. Overdue channel participants should also be cleared regularly to reduce unnecessary expenses.
- **Multi-level supervision.** The proposed scheme should be able to ensure that the whole process from the establishment of the off-chain payment channel to the termination of the transaction process is effectively regulated. The identity information of the transaction participants and the specific process of the transaction should be supervised by the supervisory users, and the dishonest behavior of the supervisory users also needs to be balanced in some way. Smart contracts on-chain should also be involved in the whole supervision process in order to achieve comprehensive multi-level supervision.

4. Multi-party payment channel supervisable scheme

In this section, we first describe the overview of our scheme. Then, we provide a detailed description of the transaction process and the supervision mechanisms of the scheme.

4.1. Overview

Our scheme focuses on the intra-channel transaction process. The case of unsupervised transaction channels directly established by both trading users and payment channel networks composed of several channels is not in the scope of this paper.

The multi-party off-chain payment channel scheme is designed as a supervisable in-line iterative process that allows trading users to transact in parallel, as shown in Fig. 2, with each round consisting of four stages, Channel Preparation stage, Transaction stage, Transaction Settlement stage and Channel Update stage. To better illustrate the relationship between the stages, only part of the process of the two rounds (i.e., round i and round $i+1$) of the scheme is shown in Fig. 2. In order to describe our scheme more clearly, we will elaborate our scheme from both transaction and supervision aspects, and the smart contracts involved in the scheme will introduce simultaneously.

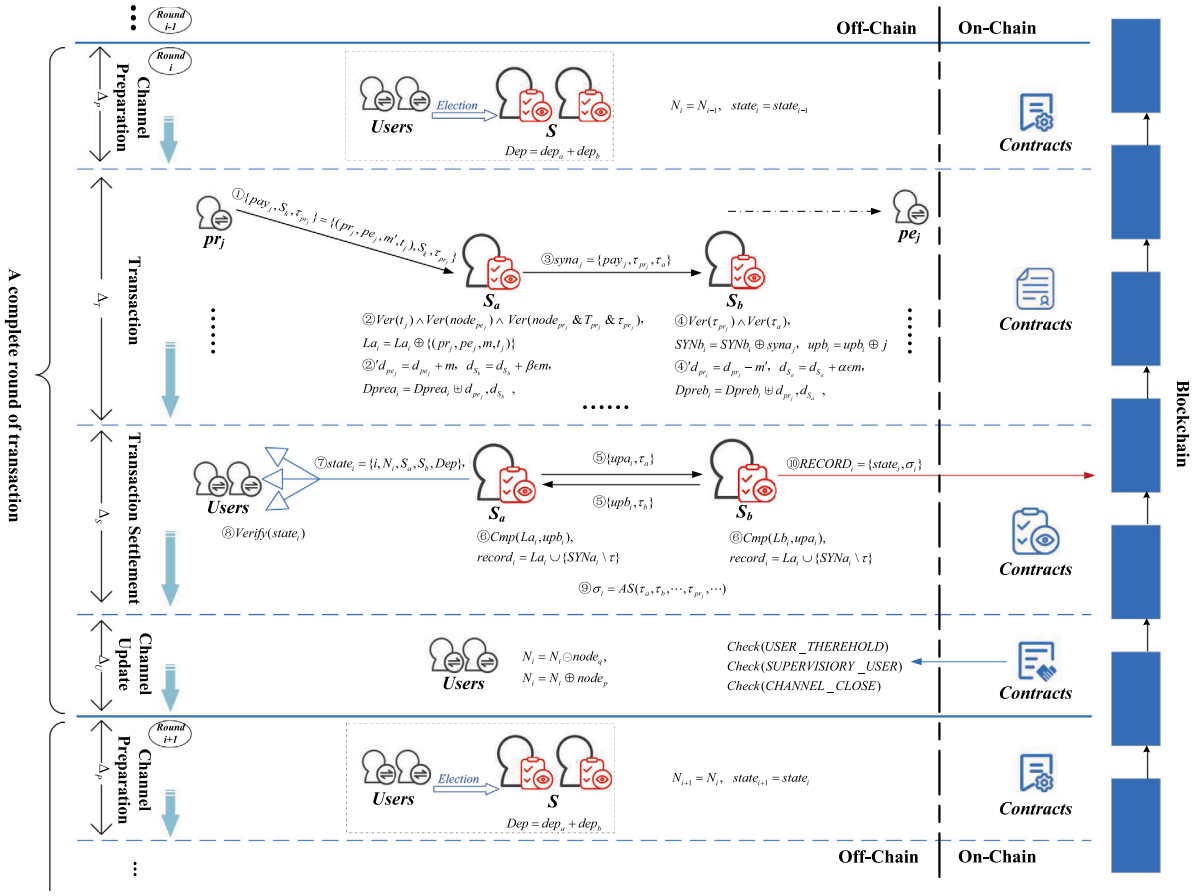

Fig. 2. The process of the four stages of the multi-party off-chain payment channel scheme, the Channel Preparation stage, the Transaction stage, the Transaction Settlement stage and the Channel Update stage, is cyclically operation and supervised.

4.2. Transaction process in channel

In our scheme, the trading users of the chain can establish a multi-party off-chain payment channel and trade in the channel in parallel, and the users in the payment channel will supervise the transaction process by themselves. Specifically, at the startup of the channel, users with trading needs trigger Channel Establishment Contract to establish the multi-party off-chain payment channel.

Algorithm 1 Channel Establishment Contract

Input: NodeGroup $NODE$, AddressGroup $ADDRESS$, PreDeposit PRE, S_a, S_b
Output: UserGroup N_0
1: **for** each $node_p$ in $NODE$, $1 \le p \le n$ **do**
2: **if** $node_p$ have passed SupervisionInformation Contract **then**
3: Get $pre_p \to d_p$ from $address_p$, $pre_p \in PRE$, $d_p \in D$, $address_p \in ADDRESS$;
4: Set $T_p = 0$;
5: **end if**
6: **return** $node_p = (id_p, d_p, T_p)$;
7: **end for**
8: **return** $N_0 = (node_1, node_2, \cdots, node_p, \cdots)$;

4.2.1. Channel preparation stage

This stage is to prepare for the subsequent transactions by renewing the basic state of the channel. Our scheme is designed to have the supervisory user act as the bookkeeper and supervisor of the transaction in the channel. If the supervisory users from the previous round still meet the conditions for continued supervision, there is no need to re-elect them. If necessary, two supervisory users are elected to supervise the transactions in the channel. Details of the election can be found in Section 4.3.

In addition, this stage assumes the responsibility of renewing the user groups as well as the channel state. Specifically, for the round i transaction process, this stage requires removing the users who exited in the previous round and adding the new users who joined in this round, thus obtaining a new user group N_i. And further renewing the channel state $state_i$ further based on the change of supervisory users. Channel Preparation Contract is shown in Algorithm 2.

Algorithm 2 Channel Preparation Contract

Input: PaymentChannel PC, ChannelState $state_i$, UserGroup N_i, CurrentRound i
Output: UserGroup N_{i+1}, ChannelState $state_{i+1}$, PaymentChannel PC
1: Get supervisory users S_a, S_b and deposit Dep;
2: **return** $N_{i+1} = N_i$;
3: **return** $state_{i+1} = (i+1, N_{i+1}, S_a, S_b, Dep)$;
4: **return** $PC = (num, state_{i+1}, \Omega)$;

4.2.2. Transaction stage

In this stage, users can initiate transaction requests to supervisory users in parallel to complete the payment process. Take the transaction j in the course of the round i of transactions as an example. Payer pr_j initiate a transaction request $\{pay_j, S_k, \tau_{pr_j}\} = \{(pr_j, pe_j, m', t_j), S_k, \tau_{pr_j}(pay_j)\}, 1 \le j \le tn, k \in \{a, b\}$ to a supervisory user S_k at t_j, which can be any time within a period of time in Transaction stage, shown

as ① in Fig. 2. pe_j is the payee of the transaction. S_k stands for the supervisory user of this transaction. $\tau_{pr_j}(pay_j)$ is the signature of pr_j on the effectiveness of the transaction pay_j. Also, since our supervisory mechanism is designed with dual supervisory users, the transaction request can be sent to any of the supervisory users. Suppose $k = a$ in this transaction. After receiving a transaction request from pr_j, S_a first validates the authenticity and legitimacy of the request. Then, the account balances of relevant users are pre-changed as follow:

$$\begin{cases} d_{pe_j} = d_{pe_j} + m \\ d_{S_b} = d_{S_b} + \beta \epsilon m. \end{cases} \tag{3}$$

In (3), $m = \frac{1+\epsilon}{m'}$ is the real amount pr_j wants to transfer to pe_j, and ϵ and β are the proportion of processing fees of overall and auxiliary supervision fees agreed in advance. Then, channel pre-change account balance $Dprea_i = Dprea_i \uplus d_{pe_j}, d_{S_b}$.

In the meantime, the information is synchronized to another supervisory user, S_b. The synchronous message received by S_b is $syna_j = \{(pr_j, pe_j, m', t_j), \tau_{pr_j}(pay_j), \tau_a(pay_j)\}$, containing both the signature τ_{pr_j} of pr_j and the signature τ_a of S_a on the effectiveness of the transaction pay_j. After receiving the message, the supervisory user S_b also verifies and then proceeds to complete the pre-change of the remaining account balance:

$$\begin{cases} d_{pr_j} = d_{pr_j} - m' \\ d_{S_a} = d_{S_a} + \alpha \epsilon m \end{cases} \tag{4}$$

In (4), α is the proportion of processing fees of main supervision fees agreed in advance. Besides, $\alpha + \beta = 1$. Then, channel pre-change account balance $Dpreb_i = Dpreb_i \uplus d_{pr_j}, d_{S_a}$. Then, the channel pre-change account balance of S_a and S_b will aggregate in the Transaction Settlement stage to finish the transaction in this round. The relevant transaction contract is shown in Algorithm 3.

Algorithm 3 Transaction Contract

Input: PaymentChannel PC, PrechangedBalance $Dprea_i$, $Dpreb_i$,
Output: AccountBalance $D_i^{'}$
1: $Dpre_i = Dprea_i \cup Dpreb_i$
2: **for** Each $dpre_p \in Dpre_i$ **do**
3: Set $d_p = dpre_p$, $d_p \in node_p, node_p \in N_i$;
4: **end for**
5: **return** $D_i^{'} = Dpre_i$;

4.2.3. Transaction settlement stage

When a round of transactions is completed, data synchronization and aggregation between supervisory users begin. Transaction requests sent at this stage will be considered invalid. Each supervisory user synchronizes the update list to the other supervisory user, and consolidates the synchronized transaction list with its own main transactions list.

$$record_i = La_i \cup SYNa_i = Lb_i \cup SYNb_i \tag{5}$$

$record_i$ contains the whole transactions that happen in round i, whereas the main transactions list is processed by one supervisory user himself combined with the synchronized transaction list processed by the other supervisory user exactly include. Then, the status of the channel $state_i = \{i, N_i, S_a, S_b, Dep\}$ is also broadcasted to all users in the channel at the end of the stage for verification. After waiting for a period of time, the supervisory users will aggregate the signature $\sigma_i = AS(\tau_a(state_i), \tau_b(state_i), \dots, \tau_{pr_j}(pay_j), \dots)$ and upload the packaged channel state and aggregated signatures during the current round of transactions to the blockchain.

$$RECORD_i = \{state_i, \sigma_i\} \tag{6}$$

4.2.4. Channel update stage

In the Channel Update stage, new channel participants can choose to join in. Users in the channel whose channel thresholds exceeded the maximum limit in this round will be forced to withdraw, and users who did not exceed can also withdraw voluntarily.

At the same time, it is necessary to judge whether the existing supervisory users satisfy the supervisory conditions, and if they do not satisfy or the supervisory users choose to exit the channel, the Supervisory Users Election algorithm is re-executed and new supervisory users are elected in the next stage, i.e, the new Channel Preparation stage in the new round.

In addition, as the last stage of the transaction process of a round, this stage also bears the responsibility of determining whether the channel should be closed or not. If the channel existence condition is not met (i.e., too few channel participants), the channel should be closed. The user account balance in the channel will be refunded to the user blockchain address by the Channel Close Contract, shown in Algorithm 5, and the deposit of the supervisory user will be refunded together.

Algorithm 4 Channel Update Contract

Input: ChannelState $state_i$, NewNode $NNODE$, AddressGroup $ADDRESS$, PreDeposit PRE, UserGroup N_i
Output: UserGroup N_i
1: **for** each $node_q \in N_i$, $1 \le q \le n$ **do**
2: **if** $T_q \ge \Omega$ or EXIT_VOLUNTARILY **then**
3: $d_q \rightarrow address_q$;
4: EXIT_CHANNEL;
5: **end if**
6: $T_q = T_q + 1$;
7: **end for**
8: **for** each $nnode_p$ in $NNODE$, $p \ge 1$ **do**
9: **if** $nnode_p$ have passed SupervisionInformation Contract **then**
10: Get $pre_p \rightarrow d_p$ from $address_p$, $pre_p \in PRE$, $d_p \in D$, $address_p \in ADDRESS$;
11: Set $T_p = 0$;
12: **end if**
13: **return** $nnode_p = (id_p, d_p, T_p)$;
14: **end for**
15: **return** $N_i = N_i \oplus (nnode_1, nnode_2, \cdots, nnode_p, \cdots)$;

Algorithm 5 Channel Close Contract

Input: PaymentChannel PC, ChannelState $state$, AddressGroup $ADDRESS$
Output: ∅
1: **for** each $node_j$ in N, $j \ge 1$ **do**
2: $d_j \rightarrow address_j$, $d_j \in node_j$, $address_j \in ADDRESS$;
3: EXIT_CHANNEL;
4: **end for**
5: CHANNEL_CLOSE;

During a new Channel Preparation stage, in addition to the probable election of supervisory users, updates of the channel state and related data within the channel are also performed. The off-chain payment channel will continue to operate until the channel is closed.

4.3. Supervision of transaction

4.3.1. Supervision architecture

In order to more comprehensively and perfectly supervise the whole process of transaction in the off-chain payment channel, a multi-level architecture of supervision is adopted in our scheme as Fig. 3, supervising ordinary trading users and supervisory users in the whole process.

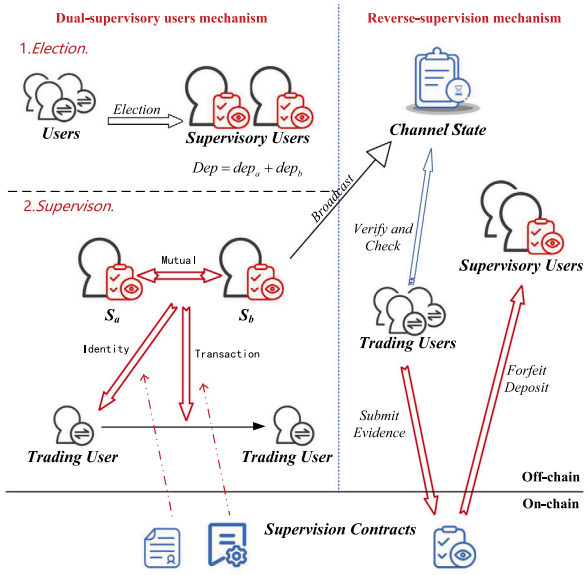

Fig. 3. Supervision Architecture.

The supervisory mechanism adopted in this scheme can be divided into two layers: on-chain and off-chain. The specific supervision mechanism design mainly focuses on the off-chain part, including the dual-supervisory users mechanism and the reverse-supervision mechanism. In the following, we will introduce them separately.

4.3.2. Supervisory mechanism

Dual-supervisory users mechanism. The dual-supervisory users mechanism, as an important part of the overall multi-party chain off-chain payment channel supervision scheme, assumes a crucial role in maintaining the normal operation of the channel. The following will introduce its operation principle in terms of election and operation.

Election. In the Channel Preparation stage, two supervisory users are to be elected from the channel participants. The election needs to satisfy the following condition: the number of users in the channel is more than three. The reason for the need to satisfy the condition is that if the number of users in the channel is less than this number, i.e., only two or three trading users in the channel, there is no need to establish a multi-party channel or the supervisory mechanism with two supervisory users proposed in this paper is not applicable.

The elected supervisory users must also satisfy a few requirements: First, the account balance of the supervisory users S_a and S_b need to satisfy $d_k \geq \frac{|N| * \{d_p\}_{max}}{2}$, $k \in \{a, b\}$, $p \in [1, n] \backslash \{a, b\}$. Second, the channel threshold T_i of supervisory users should be less than Ω. The reason for the first requirement is that paying a certain deposit can effectively avoid supervisory users from behaving maliciously, and other users in the channel can forfeit the deposit in case of supervisory users' improper supervision. Details will be described in Section 4.3. And the second requirement is to avoid the situation that one trading user without needs or already offline still stays in the payment channel causing unnecessary waste. The Supervisory Users Election algorithm is shown in Algorithm 6.

Due to the importance of the supervisory users in the transaction process, the scheme designs corresponding supervisory measures to ensure the fairness and legality of the election process. When a Channel Preparation Contract is triggered, the SupervisionInformation Contract is triggered together to verify whether the user's account balance in the channel meets the conditions. When the supervisory user election

Algorithm 6 Supervisory Users Election

Input: ApplicantGroup P, AccountBalance D, ChannelThreshold T
Output: The supervisory users S_a, S_b and deposit Dep

```
for each P_i ∈ P, 1 ≤ i ≤ n do
    if T_i ≥ Ω then
        ILLEGAL_APPLICANT;
    end if
    if d_i ≤ (|N| * {d_p}_max)/2, p ∈ [1,n]\{i} then
        INSUFFICI_ACCOUNT_BALANCE;
    end if
    Add P_i to S;
end for
Select the two with the lowest T in S as the supervisory users of this round of transaction;
return S_a, S_b
d_a = d_a − dep_a, d_b = d_b − dep_b ;
return Dep = dep_a + dep_b ;
```

algorithm is executed, there are also corresponding supervisory contracts to determine the election priority to ensure the legitimacy of the supervisory user.

Operation. In the Transaction stage, supervisory users actively supervise the transaction process by verifying the transaction request or information from other users. Just as ② and ④ in Fig. 2. For the supervisory user S_a who received the transaction request, he can supervise transactions from multiple perspectives, such as transaction time, trading users, and transaction content and the process of verification is as follows.

$$\begin{aligned} &Ver(t_j \in [t_i + \Delta_P,\ t_i + \Delta_P + \Delta_T]) \wedge \\ &Ver(node_{pe_j} \in N_i) \wedge \\ &Ver(node_{pr_j} \in N_i \& T_{pr_j} < \Omega) \& \tau_{pr_j}(pay_j). \end{aligned} \tag{7}$$

Specifically, when supervisory user S_a receives a transaction request, he first verifies whether the request was sent within the current round of transactions, i.e. $t_j \in [t_i + \Delta_P,\ t_i + \Delta_P + \Delta_T]$, to ensure the legitimacy of the transaction request. Verifying the transaction request time confirms whether the transaction is part of the current transaction process and prevents the attacker from sending a previous legitimate transaction to interfere with the current transaction. In addition, to ensure that transactions are performed sequentially, transactions that contain a timestamp much smaller than the time of the most current transaction will also not pass verification. Then, to ensure the legitimacy of the payer, S_a further verifies if the payee pe_j and payer pr_j belongs to the current round of channels. The purpose of verifying the payee is to ensure that the transaction process is achievable, and the purpose of verifying the payer is to prevent malicious attacks. Next, S_a checks whether the payer's channel threshold T_{pr_j} has exceeded the maximum requirement Ω or not to further make sure the legitimacy of the user initiating a transaction in the channel. Finally, S_a proceeds to verify the signature $\tau_{pr_j}(pay_j)$ of the payer pr_j to ensure the authenticity of the transaction request, which also ensures the non-repudiation of the transaction by the payer. If any of the above verification processes fail, S_a will discard the request. After verifying that the request is correct, S_a adds the request (pr_j, pe_j, m, t_j) to the main transaction list La_i of S_a for round i.

$$La_i \oplus \{(pr_j, pe_j, m, t_j)\} \tag{8}$$

As shown in Fig. 2, the other supervisory user S_b, after receiving the synchronized message, also has to perform the following verification to determine the effectiveness of the transaction.

$$Ver(\tau_{pr_j}(pay_j)) \wedge Ver(\tau_a(pay_j)) \tag{9}$$

Specifically, the secondary verification of the payer's signature is to prevent its collusion with the supervisory user S_a, and the verification of the supervisory user's signature is to prevent the forgery of the synchronized message. Once pr_j or S_a's signature is found to be faulty or otherwise disruptive to normal transactions, S_b immediately submits this as evidence to the blockchain to trigger the corresponding supervisory contracts and punish the relevant channel participants. If both signatures pass verification, S_b adds the authenticated synchronized information to the list $SYNb_i$ and records the corresponding transaction serial number j.

$$SYNb_i = SYNb_i \oplus syna_j, upb_i = upb_i \oplus j \tag{10}$$

In the Transaction Settlement stage, supervisory users continue to take on the role of overseeing the performance of the other supervisory user. In order to consolidate all the transactions that occurred in this round, supervisory user S_a sends $\{upa_i, \tau_a(upa_i)\}$ to S_b. Correspondingly, S_b sends $\{upb_i, \tau_b(upb_i)\}$ back. The update list is to further verify the supervision process and to prevent certain dishonest actions by the supervisory users. Specifically, just as ⑥ in Fig. 2, supervisory user S_a, upon receiving the update transaction list upb_i, compares the transaction serial numbers therein with its recorded main transaction list La_i to determine whether supervisory user S_b dishonestly ignores the received synchronized transaction message $syna_j$ or maliciously adds fictitious transactions to reach illegal rewards. Since the supervisory user signs this message, it cannot be denied, i.e. the subsequent addition of false forged transactions is not valid. If there is inconsistency, the record will be submitted to the blockchain to call for punishment. This stage will last for δ time, and if the supervisory users still have not submitted the evil judgment at the end of the time, the supervisory users will approve the current round of transactions by default.

In the Channel Update stage, supervisory users are responsible for determining whether a channel participant meets the conditions to continue a transaction in the channel, i.e., they check trading users' channel threshold $T_p < \Omega$ or not. For users who do not meet the conditions, Channel Update Contract is triggered to make them exit from the channel. Note that the channel thresholds of supervisory users are determined by the Supervisory User Election algorithm.

Reverse-supervision mechanism. The reverse-supervision mechanism in the multi-party under-chain payment channel supervised scheme is also one of the keys of the scheme. The reverse-supervision mechanism for ordinary trading users means that the supervisory user's power within the channel is checked and the high authority given to the supervisory users by the scheme is bounded. This contributes to the healthy functioning of the channel.

Specifically, trading users reverse-supervision the behavior of supervisory users by checking the information they broadcast. After receiving the broadcast of the channel state, trading users in the payment channel can verify the channel state $state_i$ to check their account balance, further verifying whether the supervisory users have honestly recorded the transactions that occurred during the current round. At this point, two supervisory users share the responsibility of recognizing the current round of channel state. Once a trading user verifies that the channel state is wrong, the supervisory contract SupervisionTransaction can be triggered, forfeiting the supervisory user's deposit and rewarding it to the channel participants.

4.3.3. Supervisory contracts

Multiple smart contracts are designed in this scheme to implement the supervision of the transaction process within the entire multi-party off-chain payment channel. We introduce the two most critical contracts in the supervisory process to enable penetrating supervision of the transaction process. Contracts are described in Algorithm 7 and Algorithm 8.

Whenever trading users want to establish an off-chain payment channel, a SupervisionInformation contract will be triggered to verify the users' identity information and the legitimacy of the address to ensure the legitimacy of the user's identity participating in the payment channel.

Algorithm 7 SupervisionInformation

Input: NodeGroup $NODE$, AddressGroup $ADDRESS$, NodeAccount NA;
Output: Supervisory information
1: **for** Each users in $NODE$ **do**
2: Check the user history;
3: **if** $id_{node} \neq address.getid()$ **then**
4: WRONG_USER_INFORMATION;
5: **end if**
6: **end for**

The other contract is the SupervisionTransaction contract, which is mainly responsible for verifying the content of each round of transactions submitted by the supervisory users, and judging and punishing any misdeeds (mainly of the supervisory user) during the transaction. As shown in Algorithm 8, the contract includes both active supervisory mechanism (lines 1–7) and passively triggered supervisory mechanism (lines 8–13).

Algorithm 8 SupervisionTransaction

Input: PaymentChannel PC, ChannelRecord $RECORD_i$, AccountBalance D_i;
Output: Supervisiory information;
1: Vereify the supervisory users' signature $\tau_a, \tau_b \in \sigma_i$;
2: **if** $Sum_{D_i} \neq ChannelBalance$ **then**
3: SUPERVISION_FAILURE;
4: **end if**
5: **for** Each $pay_j \in record_i$ **do**
6: Verify the users' signature $\tau_{pr_j} \in \sigma_i$;
7: **end for**
8: **if** $getProof() == SupMsg$ **then**
9: Check the other supervisory user based on the proof;
10: **end if**
11: **if** $getProof() == UserMsg$ **then**
12: Check the supervisory users based on the proof;
13: **end if**

5. Security analysis

The security of the off-chain payment channel transaction scheme described in this paper is mainly ensured by the security of the supervision mechanism. For the security of the whole off-chain payment channel scheme, we give the following theorem:

Theorem 1. *If the on-chain and off-chain multi-level security supervision mechanism designed in the scheme is proofed to be secure, then the multiparty off-chain payment channel scheme we designed achieves security.*

As described in Sections 4.2 and 4.3, several smart contracts are designed in this paper, including both five basic smart contracts for maintaining channel transactions and two contracts for implementing supervision of the transaction process. From the perspective of process integrity, we can prove that the on-chain smart contract design is complete so that the on-chain part of the multi-level security supervision mechanism achieves security.

5.1. Analysis of supervisory users

The supervisory users, as important entities elected in the channel to maintain the stable performance of the channel, are responsible for verifying the signature and the timestamp in each transaction initiation request to prevent dishonest behavior of the trading users

from disrupting the normal running of the transaction. The ability of users to conduct normal transactions in the off-chain payment channel relies heavily on the effective action of supervisory users. We will first give a theorem and then give a proof for each of the three items of Definition 2.

Theorem 2. *If the dual supervisory users in the off-chain payment channel are incentivized to properly exercise their supervision responsibilities, while their authority is constrained so that they cannot bear the consequences of malicious behavior, i.e, Bhv_S is satisfied, the positive supervision part of the off-chain security supervision scheme is considered to be complete.*

Proof. $Need_S$: When a trading user wants to initiate multiple transactions in a short period of time for the purpose of double spend attacks to achieve fraud, the supervisory user verifies the time stamp of the initiated transaction request and the real-time account balance of the user to determine whether the trading user satisfies the conditions of the transaction, and rejects the transaction if it does not. Also, since the transaction request contains a signature, which means that the user confirms the details of the transaction, the user is non-repudiation. By the same logic, other attacks such as replay attacks by the user are also ineffective(i.e., multiple identical transaction requests will be distinguished into multiple transactions by the supervisory users based on the timestamp). □

In order to incentivize supervisory users to actively participate in transactions, our scheme is designed to include processing fees for supervisory users in each transaction. This means that for transactions that happened in a normally operating channel, the more the transactions happen, the more amount of fee revenue supervisory users achieve. When the normal operation of the channel is disrupted, the supervisory user's fee revenue disappears instantly. Therefore, a rational supervisory user will only actively supervise each transaction.

Proof. $Incentive_S$: The ratio ϵ of the processing fee for every transaction and the specific allocation ratio α and β ($\alpha + \beta = 1$) are agreed in advance. In addition, since the pre-changers of the channel account balance, including the processing fee ϵm, are supervisory users. In order to reduce the relevant risk, avoiding the direct manipulation of their own account, the amount is deducted or increased by the other supervisory users. Specifically, supervisory user S_a, who receives the pr_j's transaction request to transfer m (m' in pay_j, containing processing fee ϵm, i.e, $m' = (1+\epsilon)m$) to pe_j, is responsible for adding the balance of another supervisory user S_b by an agreed percentage $\beta\epsilon m$; another supervisory user S_b is responsible for adding the supervisory user S_a's channel account balance by $\alpha\epsilon m$. Transactions within the channel can now be conducted in an orderly manner under supervision. □

In the previous supervision schemes for payment channels, there was generally only a single high authority supervisory mechanism with a presumption of trustworthiness. However, the reality is that dishonest acts committed by higher authorities are more harmful to the channel. Therefore, how to check and balance the high authority supervisory mechanism is an urgent issue to be solved. This scheme uses dual supervisory users to jointly supervise the transactions in the channel, and both supervisory users have the same authority. Due to their own competitive relationship, they cannot form complicity, but will instead constrain each other and prevent dishonest behaviors.

Proof. $Constraint_U$: During the Transaction Settlement stage of a round of transactions, as messages need to be synchronized between the dual supervisory users, which in turn consolidate the state of the current round of transactions, one supervisory user can supervise the messages sent or broadcast by the other supervisory user. Just as ⑥ in Fig. 2, supervisory user S_a can compare the details in message upb_i to check whether the other supervisory user S_b record the synchronized messages honestly. This is the same for messages broadcasted out by himself. Once the negligence of the supervisory user S_a is discovered, the supervisory user S_a can request the blockchain to impose a penalty on the S_b. □

5.2. Analysis of trading users

In our scheme, trading users also assume the responsibility of monitoring the supervisory users. In other words, during the process of transaction in the channel, the channel participants can also monitor the supervisory users at all times, and once the dishonest behaviors of the supervisory users are found, the evidence can be submitted to the blockchain to trigger the SupervisionTransaction Contract and forfeit the supervisory users' margin. We will first give a theorem and, as $Constraint_U$ for trading users is the same as the $Need_S$ of supervisory users proved above, then give a proof for each of the other two items of Definition 1.

Theorem 3. *If the ordinary trading users in the off-chain payment channel are restrained from obtaining improper benefits through malicious behaviors, and at the same time ensure the smooth execution of transactions by supervising the behavior of supervisory users, i.e, Bhv_U is satisfied, the reverse supervision part of the off-chain security supervision mechanism is considered to be complete.*

Proof. $Need_U$: The trading users can verify the channel state information broadcast by the supervisory users to determine whether any supervisory user has honestly recorded each transaction, added or deducted account balance. during the current round of transactions, thereby inferring whether the supervisory users have committed mischief. If supervisory users are found to have performed each operation within the round in good faith during the given verification period, it can be determined that the supervisory user is not at fault and it also constitutes an acceptance that the fees earned by supervisory users during the round are legitimate. Without evidence of a supervisory user's dishonest behavior, a channel participant would not be able to trigger Supervisory Contract, and therefore would not be able to deduct the supervisory user's margin maliciously. □

Proof. $Incentive_U$: In the event of forfeiting the supervisory user's margin, the evidence submitter receives half of the deposit of the default supervisory user, and the remaining deposit is divided equally among the other participants in the round. Through such a means to motivate channel participants to actively participate in the monitoring process of transactions in the channel, which helps to further ensure the safety and reliability of the channel. □

5.3. Collusion in payment channel

As there are two types of users, trading users and supervisory users, in our scheme, the possible collusion within the channel is of three kinds, namely, collusion among trading users, collusion among supervisory users, and collusion between a supervisory user and a trading user. We assume that the proportion of honest users in the channel is not less than 50%, so the complicity of ordinary transaction users cannot interfere with the normal transaction process.

The supervisory user role set up in the scheme has powers and obligations beyond those of normal users, i.e. they need to carefully verify every transaction in the payment channel, keep detailed records and update user balances in the channel, and consolidate the information to the blockchain at the end of each round of the channel. If the supervisory users choose to collude in order to gain improper benefits, the channel will not be able to operate safely and securely. In this scheme, two supervised users in the channel are required to jointly pay a deposit of $|N| * \{d_j\}_{max}$ amount and their account balance is required to be greater than $\frac{|N| * \{d_j\}_{max}}{2}$ before they are elected as supervisory users. In the process of transaction supervision, the two supervisory users are in competition with each other. When one of them commits evil, the other one can submit the evidence of the evil of the supervisory user to the blockchain to trigger the supervisory

Table 1
Comparison of transaction and supervisory effects.

Scheme	Number of users in the same channel	Multi-party transactions	Dynamic fund in channel	Number of supervisor	Supervisor from internal or external	Active or Passive supervision	Other supervisory method in channel
LN [3]	2	×	×	0	–	–	–
Pisa [25]	2	×	×	1	External	Active	Signed receipt
Scheme in [26]	2	×	×	1	External[c]	Active	Anti-collusion contracts
Scheme in [14]	2	✓	×	0	–	–	–
MPC [16]	N[a]	✓	×	0	–	–	–
Gnocchi [17]	N	✓	×	1	Internal	Active	Fraud proof mechanism
Scheme in [18]	N	✓	×	1	Internal	Passive	–
Magma [27]	N	✓	×	1	External[d]	Active-Passive	Reports
Our Scheme	N[b]	✓	✓	2	Internal	Active-Passive	Reverse-supervision, supervisory contracts

[a] This scheme is designed to accommodate an almost UNLIMITED number of nodes in the channel.
[b] We introduce channel threshold to LIMIT the number.
[c] Watchtower.
[d] Operator.

smart contract and forfeit the supervisory user's margin to the trading participants in the channel. The revenue gained in this process is much greater than the revenue generated by the collusion. However, if both supervisory users behave honestly, supervisory users are also unable to maliciously submit evidence of the other's misdeeds to gain additional revenue. Therefore, the problem of supervisory users' collusion in the process of transaction supervision can be solved.

If there is a collusion between a trading user and one of the supervisory users, there may also be double spending attacks or other attacks to disrupt the normal operation of the payment channel. In our scheme, since the transaction initiate request contains the signature and timestamp of the trading initiator user, the synchronization message from one supervisory user to another also contains the signature for the supervision of every transaction request, and the signature of the supervisory users are also included in the channel state broadcast in the transaction settlement stage, it is unrealistic to deny the facts that have been signed, so it is impossible to commit collusion in a channel in which no less than half of the members are honest and upright.

5.4. Other issues

Supervisory users, as participants in the channel, can also conduct transactions normally. One supervisory user can initiate a request to another supervisory user when he wants to initiate a transaction request. Since there is no collusion between supervisory users, the process of increasing or decreasing the amount and synchronizing the messages is not affected. Therefore, supervisory users can also participate in transactions normally.

6. Evaluation

To ensure the feasibility of the proposed scheme, this thesis designs experiments from various aspects to verify the effectiveness of the multi-party off-chain payment channel scheme.

6.1. Effect analysis

To better illustrate the effectiveness of our proposed scheme, we analyze the trading effect as well as the supervisory effect of the scheme in this subsection.

6.1.1. Effect of transaction

Our scheme offers convenience, flexibility and high reliability compared to other literature on off-chain transaction processes. The comparison results are shown in Table 1. As mentioned in Section 1, traditional off-chain dual-user payment channel schemes have encountered increasing limitations due to application scenario constraints. While some schemes, such as schemes in Table 1 (Line 2–5), have attempted to enable multi-user transactions using traditional channels, they continue to face overhead challenges associated with channel establishment and termination. However, as in paper [16] (Line 6), the multi-party channel is designed to accommodate almost unlimited users, which also brings additional overhead, i.e., not all users who join the channel have transaction needs; the state of users who are offline in the channel also needs to be updated. In our proposed scheme, we introduce a channel threshold to forcibly remove expired users, thereby enhancing channel efficiency compared to existing schemes. This innovation addresses the limitations associated with traditional multi-party channel designs and represents a significant advancement in off-chain payment channel efficiency.

In addition, realizing the dynamic balance changes of user channel accounts within the channel is also a crucial indicator to judge the high availability of the channel. In traditional off-chain payment channels(Line 2–5 in Table 1), only two parties participate in the transaction in each channel, and if the balance in the channel is insufficient, the channel is closed and a new channel is created to continue the payment. However, in the multi-party channels(Line 6–9 in Table 1), such a course of action is clearly not feasible. While the rebalancing method of the payment channel network can maintain the channels to some extent, the additional consumption caused by routing and the cross-channel process is undesired. Therefore, it is critical to support dynamic balance changes within the channel. In our scheme, trading users are free to join or exit the channel in each round of transactions, i.e., they can choose to join or exit at the ChannelUpdate Stage without interfering with the normal channel operation process. Users with insufficient channel balance can choose to realize the communication between their balance on the blockchain and the channel balance in this stage to realize the dynamic balance change within the off-chain channel.

6.1.2. Effect of supervision

We qualitatively compare the supervisory effects of our scheme with those in [17,18,25,27]. The comparison compares the effectiveness of supervision from various aspects, including the number of regulatory

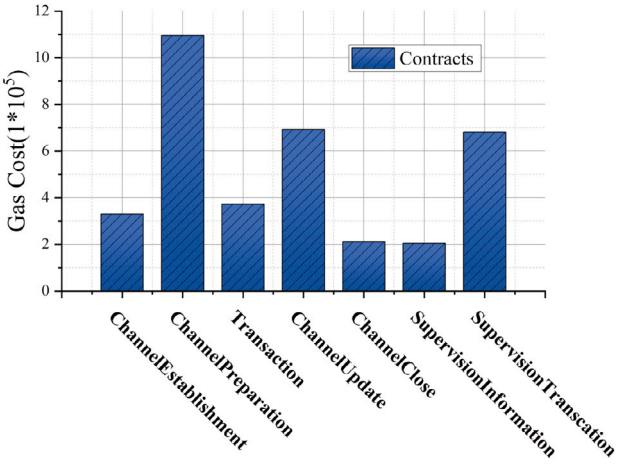

Fig. 4. Gas consumption of all smart contracts mentioned in our scheme.

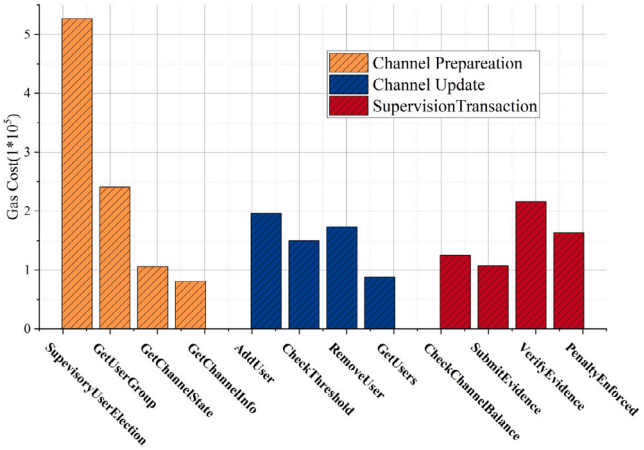

Fig. 5. Gas consumption of Channel Preparation Contract, Channel Update Contract and SupervisionTransaction Contract.

users, whether it is third-party supervision or autonomous supervision, whether the means of supervision is active or passive. The comparison results are shown in Table 1.

Scheme in [25](Line 3) and scheme in [26](Line 4) are both built on the basis of off-chain payment channel network, and they use watchtowers as external supervisors. Although this guarantees the normal operation of the channel to a certain extent, the watchtower is not involved in the actual transaction process, and can only regulate by detecting changes in the channel state. Scheme in [27] (Line 9) combines passive supervision, but the supervisors of these three schemes are kind of third-party, and their reliability is questionable. The supervisors of scheme in [17](Line 7) and scheme in [18](Line 8) are internal to the channel, but their supervisory mechanism is not well considered, and the combination of on-chain and off-chain is not taken into account. As described in Section 5, our scheme forms a comprehensive supervisory system. We set up two supervisory users within the channel, and design a supervisory mechanism that organically combines forward and reverse, active and passive, on-chain and off-chain, which is significantly better than other off-chain payment channel schemes.

6.2. Evaluation of smart contracts

We implemented the smart contracts of our scheme on Ethereum with the language of Solidity ^0.8.0 and analyzed the seven of them which are presented in Section 4 in Remix, an Ethereum-based smart contract debugging tool. In Ethereum, a certain amount of "Gas" is paid for each transaction to incentivize miners to execute the transaction and calculate the result. Gas is a unit of valuation in Ethereum that measures the resource consumption required to execute smart contracts or transfers. The Gas consumption of these seven contracts during their execution can be shown in Fig. 4.

We can clearly see that among the smart contracts corresponding to the five cycles of the off-chain payment channel operation, the ChannelPreparation Contract has the largest consumption of Gas. This is because this contract contains several processes such as supervisory user election and channel state updating. The second most consumed contract is the ChannelUpdate Contract, as it is responsible for the removal of expired users and the addition of new users. Fig. 5 shows the Gas consumed during the execution of the four main functions in the contract. The least consumed Gas consumption contract is the ChannelClose Contract, which is not among the four stages of our scheme design and has no other high consumption operations except returning the user's account balance to the blockchain address. The information shown in the diagram is the same as the original intention of our smart contract design, so it can show the feasibility of the scheme.

The Gas consumption of the two main supervisory contracts, the SupervisionInformation Contract and the SupervisionTransaction Contract, can also be seen in Fig. 4. The SupervisionInformation Contract is mainly used to verify the identity of the participating trading parties, so the amount of Gas consumed per time is very small. The SupervisionTransaction Contract, on the other hand, is responsible for the supervision of the whole trading process, and its function is both active verification of the account balance and passive supervision initiated after the user submits evidence, so the consumption of Gas is larger.

Fig. 5 also reflects the comparison of supervisory contracts with other contracts. Compared with the ChannelPreparation Contract and ChannelUpdate Contract, the two contracts with the highest Gas consumption of the transaction process, the SupervisionTransaction Contract, which fully supervises the transaction process, does not consume too much Gas. Therefore, it can further illustrate the feasibility of our design of the on-chain supervision part.

6.3. Evaluation of multiparty payment channel performance

To effectively evaluate our designed multi-party off-chain payment channel scheme, we simulated and experimented with the channel scheme in the blockchain network simulator SimBlock [28] and compared it with other kinds of schemes. To simplify the process of experimental simulation, we restrict the block size to less than 120kB, our proposed multi-party off-chain payment channel can accommodate an upper limit of 30 users, and the duration of each round of the scheme is 180 s, where $\Delta_P = 10$ s, $\Delta_T = 100$ s, $\Delta_S = 50$ s, $\Delta_U = 20$ s.

As shown in Fig. 6, we simulate the blockchain network in Simblock for channelless transactions, two-party off-chain payment channels, unsupervised multi-party payment channels, and our scheme proposed in this paper. The difference in scalability between the schemes is measured by testing the number of transactions that can be accommodated in the same size block. The experimental simulation results reveal the advantages of our scheme. At the tenth block of the blockchain (i.e., Block 9), the no-channel transaction scheme can store only about 3700 transactions on the chain, while the scheme using the off-chain payment channel technology can accommodate three times as many transactions at a minimum. This demonstrates the increased scalability of off-chain payment channels. Compared with the traditional two-party off-chain payment channel, the multi-party off-chain payment channel can accommodate more transactions in the same block because it avoids the overhead of frequent channel creation and closure. Our scheme is less scalable (about 15% less) than the unsupervised channel scheme because of the well-designed supervision mechanism, but it is still in an acceptable range.

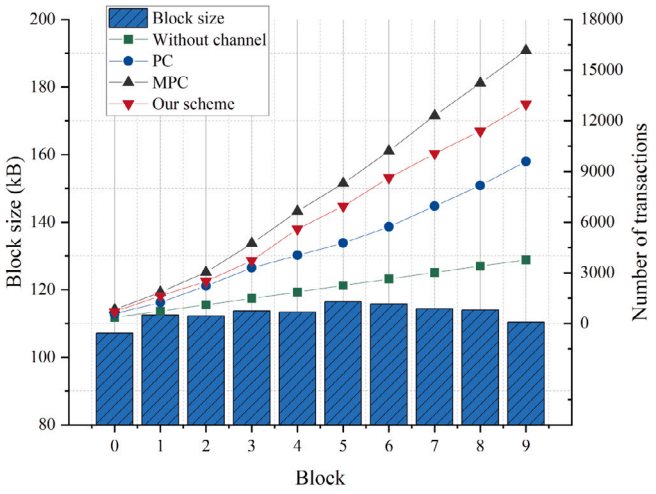

Fig. 6. Comparison of the actual number of transactions blocks accommodated under different schemes as the blocks grow.

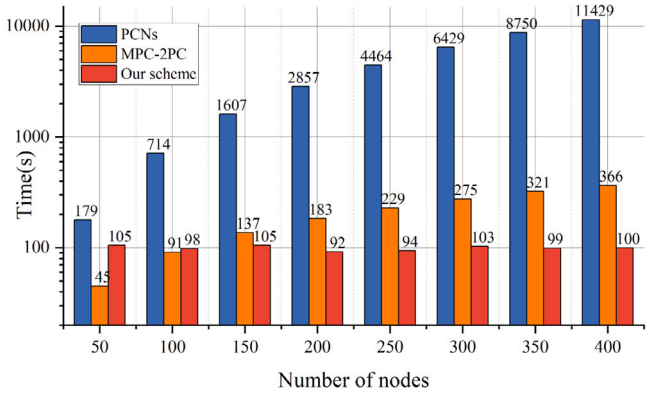

Fig. 7. As more nodes are added, the average time between the initiation of a transaction request in the channel and the posting of the transaction (assuming all users transact once).

Another advantage of our scheme is the real-time settlement without closing the channel at the end of each round of transactions. In a normal payment channel network, there is a process where the balance is locked by the channel after the transaction is closed, i.e., the settlement process will continue for a long time unless the user immediately initiates an end-of-channel request to return the balance to the blockchain address. Fig. 7 demonstrates this, and as the number of nodes participating in the channel increases, only our scheme can complete the settlement of transactions in a relatively stable amount of time.

We have also experimented with the supervisory users mechanism involved in the scheme. From Section 5 we can see that the dual-supervisory users mechanism is significantly better than the single-supervisory user or zero-supervisory user, so does it mean that more supervisory users are better? Fig. 8 shows our comparison of the dual-supervisory users mechanism and the triple-supervisory users mechanism in the Transaction stage and the Transaction Settlement stage (the overhead of the two different mechanisms in the Channel Preparation stage is negligible, and no supervised users are involved in the Channel Update stage).

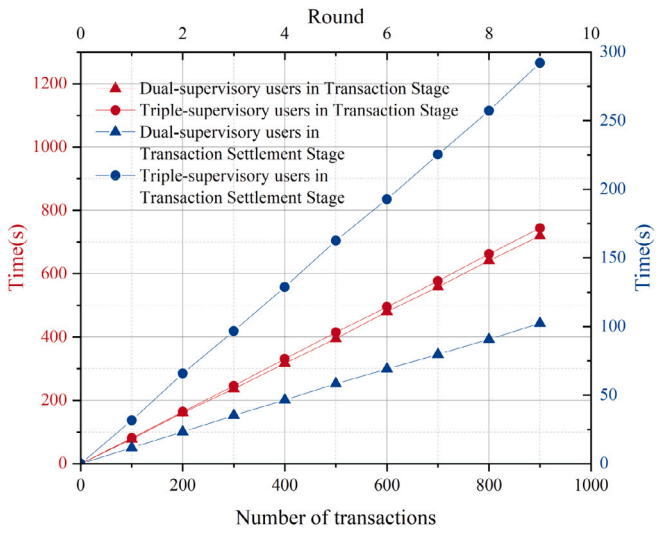

Fig. 8. Comparison of overhead from different number of supervisory users in different stages.

For the Transaction stage of our scheme (red line in the figure), the additional overhead of the dual-two supervision mechanism is about the same as the number of transactions grows, however, in the Transaction Settlement stage (blue line in the figure), the additional overhead of the triple-supervisory users mechanism is significantly more than that of the dual-supervisory users mechanism due to the synchronization process of transaction information. Even the trend of the curve shows that the number of transactions will continue to grow leading to an unacceptable overhead in the Transaction Settlement stage. Therefore, the dual-supervisory users mechanism is better than the triple-supervisory users mechanism.

7. Discussion and limitations

In this paper, we have proposed a novel off-chain payment channel scheme that supports multi-party parallel transactions, optimizes existing transaction processes, and implements a supervisory mechanism combining passive and active approaches, as well as on-chain and off-chain coordination. Through our scheme, we effectively resolve issues such as difficulties in multi-party transactions, inflexible transaction processes, and lack of transaction supervision. However, our research has certain limitations that require further investigation. Firstly, our scheme does not consider the implications of off-chain payment channel networks, which may introduce new security concerns that require additional supervision. Secondly, our supervisory mechanism has some limitations; for instance, the scheme does not support the supervision of users involving less than three parties. Further research is necessary to address these limitations and improve the effectiveness of our proposed scheme.

8. Conclusion

To address the current problems of multi-party transaction, we proposed a secure, supervisable, and concurrent off-chain payment channel scheme, where the four-stage cyclic transaction process greatly enhances the flexibility of the off-chain payment channel, the dual-supervisory users mechanism of on-chain and off-chain collaboration that intersperses the whole transaction process effectively guarantees the security of transactions, the reverse-supervision mechanism provided for ordinary trading users further guarantees the reliability of supervision. The proposed scheme has been analyzed and evaluated that with a modest overhead (about 15%), our scheme is able to

achieve secure supervision within the channel, showing its feasibility and superiority over existing approaches.

In the future, we may focus on building secure transaction channels in payment channel networks and enabling privacy-preserving secure transactions across multi-party transaction channels. By resolving the scalability issues of the blockchain and introducing a more efficient and reliable payment channel scheme, we hope to facilitate more widespread adoption of digital currency and promote the healthy development of the blockchain ecosystem.

CRediT authorship contribution statement

Ke Xiao: Conceptualization, Writing – original draft. **Jiayang Li:** Methodology, Writing – review & editing. **Yunhua He:** Writing – review & editing. **Xu Wang:** Formal analysis. **Chao Wang:** Data curation, Software.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data that has been used is confidential.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China under Grant 62272007, 61932011, 62272195 and 61802004; in part by the Beijing Natural Science Foundation, China under Grant M21029; in part by the Excellent Young Talents Project of the Beijing Municipal University Teacher Team Construction Support Plan, China under Grant BPHR202203031; in part by the National Key Research and Development Plan 2020, China under Grant 2020YFB1005600; and in part by the Guangdong Key Research and Development Plan 2020, China under Grant 2020B0101090002.

References

[1] Statista, Cryptocurrencies - Worldwide, 2023, URL https://www.statista.com/outlook/dmo/fintech/digital-assets/cryptocurrencies/worldwide.

[2] N. Papadis, L. Tassiulas, Blockchain-based payment channel networks: Challenges and recent advances, IEEE Access 8 (2020) 227596–227609, http://dx.doi.org/10.1109/ACCESS.2020.3046020.

[3] J. Poon, T. Dryja, The bitcoin lightning network: Scalable off-chain instant payments, Percept. Psychophys. 18 (2016) 205–208.

[4] Raiden Network, What is the raiden network? 2023, URL https://raiden.network/101.html.

[5] A. Gangwal, H.R. Gangavalli, A. Thirupathi, A survey of layer-two blockchain protocols, J. Netw. Comput. Appl. 209 (2023) 103539, http://dx.doi.org/10.1016/j.jnca.2022.103539.

[6] A.I. Sanka, R.C. Cheung, A systematic review of blockchain scalability: Issues, solutions, analysis and future research, J. Netw. Comput. Appl. 195 (2021) 103232, http://dx.doi.org/10.1016/j.jnca.2021.103232.

[7] P. Wongthongtham, D. Marrable, B. Abu-Salih, X. Liu, G. Morrison, Blockchain-enabled Peer-to-Peer energy trading, Comput. Electr. Eng. 94 (2021) 107299, http://dx.doi.org/10.1016/j.compeleceng.2021.107299.

[8] N. Lasla, M. Al-Ammari, M. Abdallah, M. Younis, Blockchain based trading platform for electric vehicle charging in smart cities, IEEE Open J. Intell. Transp. Syst. 1 (2020) 80–92, http://dx.doi.org/10.1109/OJITS.2020.3004870.

[9] R. Jabbar, N. Fetais, M. Kharbeche, M. Krichen, K. Barkaoui, M. Shinoy, Blockchain for the internet of vehicles: how to use blockchain to secure vehicle-to-everything (V2X) communication and payment? IEEE Sens. J. 21 (14) (2021) 15807–15823, http://dx.doi.org/10.1109/JSEN.2021.3062219.

[10] A. Miller, I. Bentov, S. Bakshi, R. Kumaresan, P. McCorry, Sprites and state channels: Payment networks that go faster than lightning, in: International Conference on Financial Cryptography and Data Security, Springer, 2019, pp. 508–526, http://dx.doi.org/10.1007/978-3-030-32101-7_30.

[11] M. Dotan, S. Tochner, A. Zohar, Y. Gilad, Twilight: A differentially private payment channel network, in: 31st USENIX Security Symposium, USENIX Security 22, USENIX Association, 2022, pp. 555–570.

[12] Y. Zhang, D. Yang, Robustpay+: Robust payment routing with approximation guarantee in blockchain-based payment channel networks, IEEE/ACM Trans. Netw. 29 (4) (2021) 1676–1686, http://dx.doi.org/10.1109/TNET.2021.3069725.

[13] Z. Hong, S. Guo, R. Zhang, P. Li, Y. Zhan, W. Chen, Cycle: Sustainable off-chain payment channel network with asynchronous rebalancing, in: 2022 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN, IEEE, 2022, pp. 41–53, http://dx.doi.org/10.1109/DSN53405.2022.00017.

[14] Y. Chen, X. Li, J. Zhang, H. Bi, Multi-party payment channel network based on smart contract, IEEE Trans. Netw. Serv. Manag. 19 (4) (2022) 4847–4857, http://dx.doi.org/10.1109/TNSM.2022.3162592.

[15] Y. Zhang, Research on multiparty payment technology based on blockchain and smart contract mechanism, J. Math. 2022 (2022) http://dx.doi.org/10.1155/2022/3434954.

[16] H. Lei, L. Huang, L. Wang, J. Chen, MPC: Multi-node payment channel for off-chain transactions, in: ICC 2022-IEEE International Conference on Communications, IEEE, 2022, pp. 4733–4738, http://dx.doi.org/10.1109/ICC45855.2022.9838626.

[17] C. Pan, S. Tang, Z. Ge, Z. Liu, Y. Long, Z. Liu, D. Gu, Gnocchi: Multiplexed payment channels for cryptocurrencies, in: Network and System Security: 13th International Conference, NSS 2019, Sapporo, Japan, December 15–18, 2019, Proceedings 13, Springer, 2019, pp. 488–503, http://dx.doi.org/10.1007/978-3-030-36938-5_30.

[18] Z. Ge, Y. Zhang, Y. Long, Z. Liu, Z. Liu, D. Gu, A high-concurrency multi-party off-chain payment scheme, Chinese J. Comput. 44 (1) (2021) 132–146, http://dx.doi.org/10.11897/SP.J.1016.2021.00132.

[19] H. Chen, Y. Lu, Y. Cheng, FileInsurer: A scalable and reliable protocol for decentralized file storage in blockchain, in: 2022 IEEE 42nd International Conference on Distributed Computing Systems, ICDCS, IEEE, 2022, pp. 168–179, http://dx.doi.org/10.1109/ICDCS54860.2022.00025.

[20] A. Singh, K. Click, R.M. Parizi, Q. Zhang, A. Dehghantanha, K.-K.R. Choo, Sidechain technologies in blockchain networks: An examination and state-of-the-art review, J. Netw. Comput. Appl. 149 (2020) 102471, http://dx.doi.org/10.1016/j.jnca.2019.102471.

[21] Z. Hong, S. Guo, E. Zhou, W. Chen, H. Huang, A. Zomaya, GriDB: Scaling blockchain database via sharding and off-chain cross-shard mechanism, Proc. VLDB Endow. 16 (7) (2023) 1685–1698, http://dx.doi.org/10.14778/3587136.3587143.

[22] L. Zhong, Q. Wu, J. Xie, J. Li, B. Qin, A secure versatile light payment system based on blockchain, Future Gener. Comput. Syst. 93 (2019) 327–337, http://dx.doi.org/10.1016/j.future.2018.10.012.

[23] T. Górski, Reconfigurable smart contracts for renewable energy exchange with re-use of verification rules, Appl. Sci. 12 (11) (2022) 5339, http://dx.doi.org/10.3390/app12115339.

[24] J. Lind, O. Naor, I. Eyal, F. Kelbert, E.G. Sirer, P. Pietzuch, Teechain: a secure payment network with asynchronous blockchain access, in: Proceedings of the 27th ACM Symposium on Operating Systems Principles, 2019, pp. 63–79, http://dx.doi.org/10.1145/3341301.3359627.

[25] P. McCorry, S. Bakshi, I. Bentov, S. Meiklejohn, A. Miller, Pisa: Arbitration outsourcing for state channels, in: Proceedings of the 1st ACM Conference on Advances in Financial Technologies, 2019, pp. 16–30, http://dx.doi.org/10.1145/3318041.3355461.

[26] M. Du, P. Yang, W. Tian, Z. Han, Anti-collusion multiparty smart contracts for distributed watchtowers in payment channel networks, IEEE J. Sel. Areas Commun. 40 (12) (2022) 3600–3614, http://dx.doi.org/10.1109/JSAC.2022.3213355.

[27] Z. Ge, Y. Zhang, Y. Long, D. Gu, Magma: Robust and flexible multi-party payment channel, IEEE Trans. Dependable Secure Comput. (99) (2023) 1–18, http://dx.doi.org/10.1109/TDSC.2023.3238332.

[28] Y. Aoki, K. Otsuki, T. Kaneko, R. Banno, K. Shudo, Simblock: A blockchain network simulator, in: IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops, INFOCOM WKSHPS, IEEE, 2019, pp. 325–329, http://dx.doi.org/10.1109/INFCOMW.2019.8845253.

Ke Xiao received the Ph.D. degree in circuits and systems from the Beijing University of Posts and Telecommunications, Beijing, China, in 2008. He has been a Professor with the North China University of Technology, China, since 2018. His research interests include IoT security and industrial Internet security. He is a member of the IEEE Communications Society and the IEEE VTS Society. He serves as a Reviewer for the IEEE Communications Letters, the IEEE Communications Magazine, and the IEEE Internet of Things Journal.

Jiayang Li received the B.S. degree from North China Electric Power University, China, in 2021. Currently, he is working toward the M.S. degree in the School of Information Science and Technology, North China University of Technology. His research interests include blockchain technology and privacy preserving.

Yunhua He received the Ph.D. degree in Computer Science from Xidian University, Xi'an, China, in 2016. He was a visiting scholar at the Department of Computer Science, the George Washington University, Washington. DC from 2014 to 2016. He has been serving as an associate Professor with the School of Information Science and Technology, North China University of Technology. His current research interests include Blockchain Technology, IoT Security and Privacy, Industrial Internet security.

Xu wang received the master's degree from the National Institute of National Hazards in 2018. She is currently working in the School of North China University of Technology. Her research interests include blockchain technology and Industrial Internet security.

Chao Wang received the Ph.D. degree from Beijing Institute of Technology, Beijing, China in 2015. He is currently an Associate Professor in North China University of Technology, Beijing, China, with Email address wangchao.andy@gmail.com. His research interests include the Internet of Vehicles and security and privacy in cyber–physical systems.