Dynamic Autonomous Cross Consortium Chain Mechanism in e-Healthcare

Rui Qiao, Xiang-Yang Luo, Si-Feng Zhu, Ao-Di Liu, Xin-Qing Yan, and Qing-Xian Wang

***Abstract*—Safe and scalable dynamic autonomous data interaction between medical institutions can increase the number of clinical trial records, which is of great significance for improving the level of medical trial collaboration, especially for clinical decision-making with regard to rare diseases. Through a preset authorization access and consensus mechanism, consortium chain provides integrity and traceability management for medical clinical data. However, how to enable users have ownership of their own medical data and share their medical data safely and dynamically between different medical institutions remains an area of particular concern. To achieve dynamic communication between medical consortium chains, this paper proposes (i) a cross-chain communication mechanism by simplifying the heterogeneous node communication topology and (ii) the construction rules of the node identity credibility path-proof to carry out dynamic construction and verification of the path-proof for cross-chain transactions. In addition, the consensus of the cross-chain transaction is modeled as a threshold digital signature process with multiple privileged subgroups; thus, the intra-chain consortium consensus based on the verification node list is extended to the cross-chain consensus. A smart contract deployment and execution scheme based on rational node value transfer mechanism is proposed by analyzing the value transfer game between nodes. Experimental results showed that the proposed scheme can not only enable patients to share their records safely and autonomously in an authorized medical consortium chain within milliseconds but also realize dynamic adaptive interaction among heterogeneous consortium chains.**

***Index Terms*—Medical collaboration, cross consortium chain, value transfer, group signature.**

Manuscript received October 1, 2019; revised December 2, 2019; accepted December 24, 2019. Date of publication January 1, 2020; date of current version August 5, 2020. This work was supported by the National Natural Science Foundation of China under Grant 61902447 and also by the Key R&D and Promotion Project of Henan Province in 2020, China (Researches on theoretical model and algorithm of performance optimization of IoT consortium chain). *(Corresponding author: Xiang-Yang Luo.)*

R. Qiao is with Zhoukou Normal University, Zhoukou, China, and also with the State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou, China (e-mail: jorui_314@126.com).

X.-Y. Luo, A.-D. Liu, and Q.-X. Wang are with the State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou China (e-mail: luoxy_ieu@sina.com; ladyexue@163.com; wqx2008@vip.sina.com).

S.-F. Zhu is with Tianjin Chengjian University, Tianjin, China (e-mail: 304854105@qq.com).

X.-Q. Yan is with the North China University of Water Resources and Electric Power, Zhengzhou, China (e-mail: yanxq@ncwu.edu.cn).

Digital Object Identifier 10.1109/JBHI.2019.2963437

I. Introduction

MEDICAL research institutions usually lack sufficient patient records for clinical trials and urgently require to access the data of other medical institutions dynamically and autonomously to increase the number of records. Thus, secure and scalable data interaction between different institutions is critical for clinical collaboration. However, there are two challenges to the dynamic autonomous interaction of medical clinical data: on the one hand, direct sharing of patient data between different medical institutions may entail privacy disclosure risks [1], [2]; on the other hand, the key requirements for medical clinical data interaction and sharing are data integrity and traceability. Medical research institutions must be able to report the reliability of their data sources to ensure the reliability of clinical trial results from data capture to final analysis. Therefore, it is of great significance to investigate the reliable sharing mechanism of e-Healthcare records for improving the effective cooperation among medical institutions.

At present, many medical institutions outsource their storage services to cloud servers. Under the centralized cloud service technology and management, the security and privacy associated with storing and sharing medical data have attracted considerable attention. For example, in [3], a solution was proposed for sharing sensitive data on the basis of non-standard diagonal data aggregation methods. In [4], a context-aware privacy protection scheme was proposed. In [5], a security model was proposed on the basis of fog computing facilities. However, the above-mentioned solutions rely on fully trusted third parties to improve the security of medical information sharing with authentication and key agreement schemes [6], which are vulnerable to offline password guessing attacks and privileged internal attacks [7]. It is difficult to achieve clinical trial data security, traceability, and management across institutions. In contrast to centralized management, a blockchain develops a chain structure to store data blocks containing the complete transaction history in a distributed manner, thus it has high tamper resistance [8], [9].

Blockchain-based data sharing mechanism can achieve integrity and traceability for medical data, so as to improve the safety and efficacy of medical data application. Therefore, the mechanism is considered as a feasible solution [10]–[12]. According to the manner of node access, blockchains can be divided into three categories: public chain, private chain and consortium chain. Under decentralized supervision, the consortium chain can implement hierarchical management of users, independent dynamic flow of values between trusted nodes, and

API-qualified queries based on the preset authorized access and consensus mechanism of the organization [13], [14]. Studies on medical consortium chains have been conducted in recent years [15]–[19]. Health Level Seven (HL7) Fast Healthcare Interoperability Resources (FHIR) was first proposed in [15]. It provides a framework for the exchange, integration, sharing, and retrieval of electronic medical information. On the basis of HL7 FHIR, a shared medical clinical data architecture FHIRChain based on Ethereum was proposed [16], and data management and access authorization in a case study of remote cancer care clinical data sharing were achieved. In [17], it was shown that smart contracts have broad application prospects in medical data mining and are expected to benefit patients on the basis of their personal medical data for research purposes. Following the above-mentioned studies, a smart contract request model based on FHIR was proposed [18]. The latest researches of blockchain in biomedical health care were summarized in [19].

Existing studies on cross-chain communication mainly focus on asset mapping and asset transaction to reconstruct the value exchange network of blockchains. Typical cross-chain technologies for asset transfer are based on hash locking and side chain technologies [20]–[23], and those for chain state transition are based on fragmentation technology [24]. With the increasing demand for asset up-chain and cross-chain communication, some studies have explored the decentralized cross-chain asset management method based on smart contracts [25]–[27]. However, the above-mentioned cross-chain communication technology mainly guarantees the atomicity of asset interaction by means of electronic cryptocurrency deployment. The logic of routing general digital assets upstream and asset interaction transaction is complex, and it cannot be directly applied to the lightweight cross-chain dynamic autonomous interaction scenario of the medical consortium chain.

At present, how to enable users to have ownership of their own medical data and share their medical data safely and dynamically between different medical institutions remains an area of particular concern. On the basis of previous work [27], a dynamic autonomous cross consortium chain mechanism for e-Healthcare data sharing based on patient privacy protection is proposed in this paper. Compared with existing medical record management methods, the proposed scheme can minimize the dependence of patients on the record generation mechanism and enable patients to selectively share their records with specific users according to their privacy preferences. Thus, the number of clinical trial records in medical institutions can be increased and the level of clinical collaboration can be enhanced.

The main contributions of this paper are as follows:

1) We propose cross-chain communication identity credibility path-proof construction rules based on the life cycle of the smart contract to achieve dynamic construction and verification of the interactive path-proof for cross-chain transactions.
2) We model the consensus of cross-chain transactions as a threshold digital signature process with multiple privileged subgroups without increasing the computational complexity. The intra-chain consortium consensus based on the verification node list is extended to the cross-chain consensus.
3) We analyze the deployment and trigger mechanism of the smart contract cross-chain, and propose the value transfer and independent interaction mechanism between nodes.

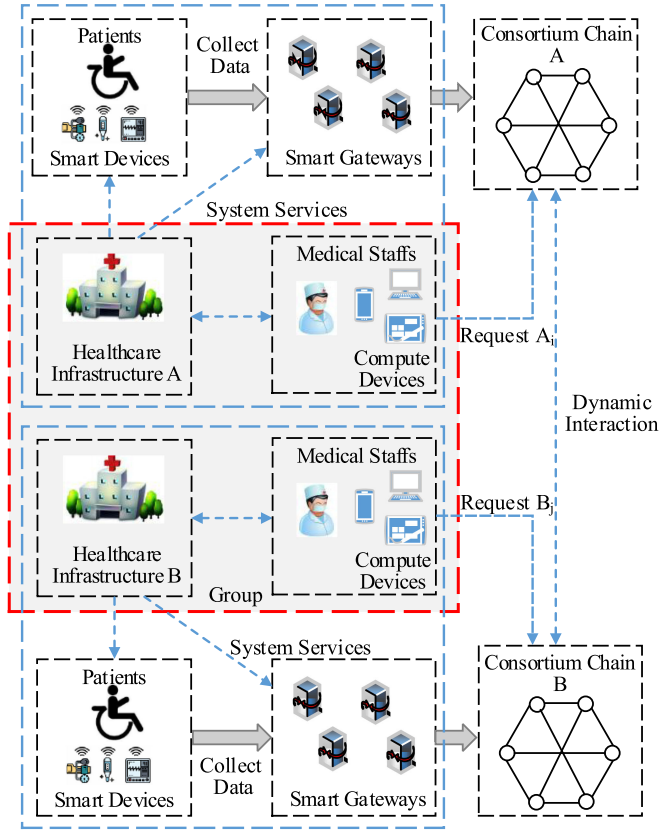

Fig. 1. System model of cross consortium chain in e-Healthcare.

The remainder of this paper is organized as follows. Section II describes the communication model between medical consortium chains and summarizes the main problems of cross-chain communication. Section III discusses the improvement of the communication mechanism between medical consortium chains from three aspects: cross-chain consensus mechanism, path-proof construction, and value transfer mechanism. Section IV proves the safety and effectiveness of the cross-chain communication mechanism through theoretical analysis and experimental deployment. Finally, Section V summarizes the study and explores directions for future work.

II. Problem Description

The system model of cross consortium chain in e-Healthcare is shown in Fig. 1. In the medical Internet of Things system, the underlying network collects user health information through medical monitoring equipment and transmits it to the convergence gateway. The convergence gateway usually has greater storage, processing, and communication capabilities than the collection device. Entities such as ordinary users, patients, and

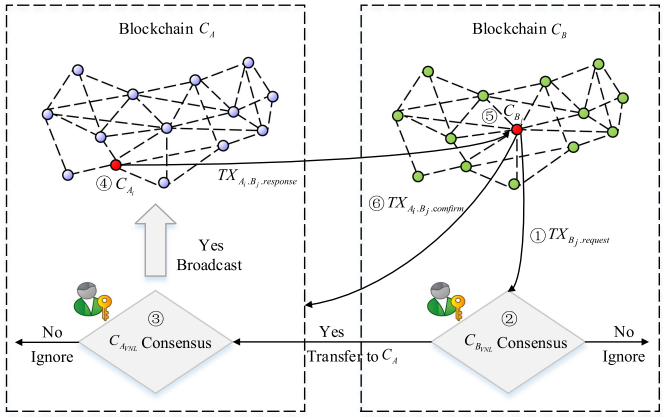

Fig. 2. Medical consortium chains communication model.

managers of the same medical institutions are common nodes in the medical consortium chain. Any node in the system can write its own data as assets to the smart contract through the gateway. The contract storage file is stored in the consortium blockchain. When the contract code receives a trigger signal from another trusted node or the smart contract, it performs corresponding operations on the storage file. Owing to the independence of the consortium chains, the data communication and value transfer between existing consortium chains in e-Healthcare still face challenges, and the problem of value isolation emerges gradually. In this paper, we introduce an efficient, secure and dynamic autonomous cross consortium chain mechanism in e-Healthcare to build healthcare IoT network to solve the above problems simultaneously.

The relevant definitions of cross-chain communications are as follows:

Definition 1: In the P2P communication mode, the consortium chain is denoted as C_X, the set of C_X is denoted as $\{C_{X_i}\}(i > 0)$, the validation node list of C_X is denoted as $\mathcal{C}_X$, and the resource owned by node C_{X_i} is denoted as asset $\xi_{C_{X_i}}$. The manner in which rational node C_{X_i} uses local asset $\xi_{C_{X_i}}$ to perform operations on message m and only output results that do not reveal private information is called blind response.

Definition 2: $\mathcal{C}_X$ is abstracted as a single point. The communication path without relay between any individual points is denoted as a single-hop path. Further, Δt represents the upper limit of time at which a node can respond to a single hop when deploying or executing a smart contract in the medical consortium chain system.

Definition 3: $\mathrm{sig}(m, X)$ represents the signature of the message $m(m \neq \phi)$ with the private key of X, the triple (m, p, σ) represents the node path-proof, $p = \{u_0, \ldots, u_k\}$ is the directed connected path from the request node u_0 to the response node u_k, and $\sigma = \mathrm{sig}(\cdots \mathrm{sig}(m, u_0), \ldots, u_k)$ is the path signature from u_0 to u_k. If u_k is a relay response node, the path-proof from u_0 to u_k is called the current path-proof. If u_k is the final response node, the path-proof from u_0 to u_k is called the full path-proof.

The cross-chain communication process is shown in Fig. 2. Suppose that C_A and C_B are cooperative consortium chains. $C_{B_j} \in C_B$ initiates an interactive request $TX_{B_j.\,\mathrm{request}}$ to $C_{A_i} \in C_A$. The cross-chain communication process consists of six stages:

1) C_{B_j} (marked with red dots in the right half of Fig. 2) constructs the identity certificate and the value transfer key $C_{B_j.s}$ and then writes the description of the interactive requirements m_{B_j} for the asset $\xi_{C_{A_i}}$, the value transfer mechanism from C_{B_j} to $\mathcal{C}_B$, and the trading deadline into the smart contract transaction $TX_{B_j.\,\mathrm{request}}$ and deploy it.
2) $\mathcal{C}_B$ verifies the transaction $TX_{B_j.\,\mathrm{request}}$ by the consensus mechanism proposed in Section III. If the verification is passed, the path-proof is updated, and the new smart contract transaction is constructed and deployed to realize the value transfer between $\mathcal{C}_B$ and $\mathcal{C}_A$; otherwise, the transaction $TX_{B_j.\,\mathrm{request}}$ is ignored.
3) If the path-proof of the transaction is verified and updated by 2), $\mathcal{C}_A$ verifies the transaction by the consensus mechanism proposed in Section III. If the verification is passed, the path-proof is updated and the smart contract is constructed and deployed to realize the value transfer between $\mathcal{C}_A$ and C_{A_i} (marked with red dots in the left half of Fig. 2); otherwise, the transaction will be ignored.
4) C_{A_i} verifies the transaction whose path-proof is updated by 3). If Equations (1)–(3) are true, C_{A_i} updates the path-proof and writes the blind response result $\mathrm{Reply}(m_{B_j})$, full path-proof, etc., into transaction $TX_{A_i.B_j.\mathrm{response}}$ and deploys $TX_{A_i.B_j.\mathrm{response}}$. Otherwise, it ignores the transaction.

$$t_{\mathrm{current}} \leq t_{\mathrm{contract\_deadline}} - \Delta t \tag{1}$$

$$\mathrm{Request}\left(m_{B_j}\right) \in \xi_{C_{A_1}} \tag{2}$$

$$\mathrm{Path}\left(m_{B_j}, p, \sigma\right) == 1 \tag{3}$$

5) C_{B_j} verifies smart contract transaction $TX_{A_i.B_j.\mathrm{response}}$ from response node C_{A_i}. After passing, C_{B_j} inputs the hash of the value transfer key $C_{B_j.s}$ to the smart contract to extract the query response $\mathrm{Reply}(m_{B_j})$. After the smart contract executes, it returns the value transfer key to C_{A_i}. The remaining stages of the value transfer smart contract are carried out in turn similarly to the above-mentioned process.
6) C_{B_j} constructs and broadcasts the transaction confirmation. Then, $\mathcal{C}_A$ verifies the confirmation from C_{B_j} and modifies the credit value of the response node C_{A_i} according to the internal incentive strategy.

In the above-mentioned cross-chain communication model, to achieve cross-chain consensus, it is necessary to establish a cross-chain consensus mechanism between the cooperating institutions on the basis of the VNL consensus mechanism. To dynamically identify a node, it is necessary to construct the path-proof of the nodes that provides the reliability identity for cross-chain communication; thus, the behavior of the nodes is limited to the credibility scope of the mechanism license. In addition, from the analysis of the honest node cooperation motivation in the medical consortium chain, it is necessary to propose a method to deploy the smart contract to realize decentralized,

dynamic, and autonomous transfer of values between the nodes of different medical consortium chains.

III. Algorithm Design

The cross-chain interaction algorithm mainly includes three parts: threshold-digital-signature-based cross-chain consensus mechanism (TCCM), path-proof construction (PPC), and value transfer mechanism (VTM). If a certain stage of the transaction processing fails, the transaction is ignored. This paper only analyzes the situation in which each stage of the transaction processing is passed.

A. Threshold-Digital-Signature-Based Cross-Chain Consensus Mechanism (TCCM)

In [27], a reputation-based consensus mechanism was proposed as follows: within the consortium blockchain, a part of the trust nodes is authorized to form VNL. The full-node server in the system is responsible for maintaining the list, and it provides all valid actions not recorded before the consensus to the verification nodes. The system relies on the verification results of VNL to reach a consensus and complete the block generation.

Based on the above-mentioned internal consensus of the consortium chain, TCCM proposed in this paper abstracts the cooperation between different consortium chains into a threshold digital group signature process and models the consensus between multiple consortium chains based on VNL for cross-chain transactions as a threshold digital signature process for multiple privileged subgroups, thereby expanding the internal consensus of the consortium chain into a cross-chain consensus among multiple consortium chains. A formal description of TCCM is given below.

Definition 4: The set of verification node lists of m consortium chains $C_1, C_2, \ldots, C_m$ that cooperate is recorded as group $\mathcal{C}$, and the verification node list of each consortium chain is recorded as m mutually disjoint privileged verification nodes subgroups (VNS) $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_m$ in group $\mathcal{C}$. Then, public-private key pair $(SK_{\mathcal{C}}, PK_{\mathcal{C}})$ of group $\mathcal{C}$ is generated on the basis of the privileged subgroup threshold signature mechanism. Thus, the cooperation relationship between the consortium chains is expressed as:

$$\mathcal{C} = \{\mathcal{C}_1 \| \mathcal{C}_2 \|, \ldots, \| \mathcal{C}_m, \mathcal{C}_i \cap \mathcal{C}_j = \phi\}, (1 \le i, j \le m) \tag{4}$$

$$|\mathcal{C}_i| = n_i, (n_i > 0) \tag{5}$$

$$\sum_{i=1}^{m} n_i = n, m \ge 1 \tag{6}$$

$$E_{SK_{\mathcal{C}_i}}(TX, t_i, n_i) = \begin{cases} \text{True, if } n_i \ge t_i' \ge t_i \\ \text{False, otherwise.} \end{cases} \tag{7}$$

$$E_{SK_{\mathcal{C}}}(TX, t_1, n_1; \cdots; t_m, n_m; t, n) = \begin{cases} \text{True, if } n_i \ge t_i' \ge t_i \\ \& \sum_{i=1}^{m} t_i \ge t \\ \text{False, otherwise} \end{cases} \tag{8}$$

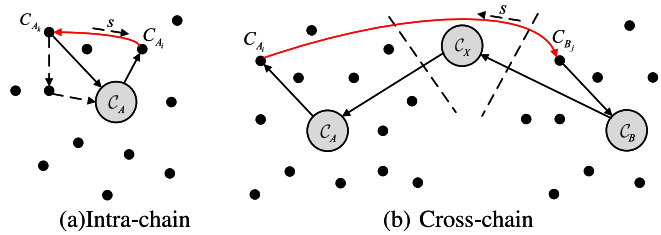

Fig. 3. Path-proof topology.

In the above-mentioned mechanism, n_i represents the number of nodes of the verification node list in subgroup $\mathcal{C}_i$, t_i represents the minimum number of n_i verification nodes in subgroup $\mathcal{C}_i$ required to pass a certain verification, and t_i' represents the actual number of n_i verification nodes in subgroup $\mathcal{C}_i$ that pass a certain verification. Further, t represents the minimum number of n verification nodes in group $\mathcal{C}$ required to pass a certain verification. Equation (4) defines the cooperation relationship between consortium chains, while Equations (5) and (6) represent the scale of the verification nodes of the cooperating consortium chains. Equation (7) represents the threshold consensus based on the verification node list in a single consortium chain, and Equation (8) represents cross-chain consensus based on the privileged subgroup threshold signature mechanism.

B. Path-Proof Construction (PPC)

The path-proof topology under P2P communication is shown in Fig. 3. The trusted propagation path is simplified on the basis of the P2P information forwarding method, and the PPC rules are as follows:

Rule 1: Under the path-proof topology, abstract the nodes of VNS as one node in the directed path-proof and correspondingly abstract the legal threshold subgroup signature (TSS) into one signature in the path signature.

Rule 2: Simplify the path-proof from one common node of the consortium chain to the VNS relayed by several common nodes to the single-hop path-proof from the node to the VNS.

Rule 3: The relay nodes in the cross-chain path-proof between different consortium chains consist of only VNS nodes.

Rule 4: The single hop in the path-proof among common nodes represents a blind response based on the value transfer key $s = C_{B_j.s}$ of the node from which the request originates.

The intra-chain path-proof topology is shown in Fig. 3(a). By Rule 1, the VNS is abstracted into a node in the directional path-proof. By Rule 2, the path-proof relayed by several common nodes from node C_{A_k} to the VNS $\mathcal{C}_A$ is simplified to the single-hop path-proof from C_{A_k} to $\mathcal{C}_A$. The cross-chain path-proof topology is shown in Fig. 3(b), where $C_{B_j} \in C_B$, $C_{A_i} \in C_A$, and $C_A \cap C_B = \phi$. By Rule 3, the relay nodes in the cross-chain path-proof from C_{B_j} to C_{A_i} consist of only VNS $\{\mathcal{C}_B, \ldots, \mathcal{C}_X, \ldots, \mathcal{C}_A\}$.

By Rule 4, the path-proofs in Fig. 3 have loops, and the path-proof between any two common nodes is a multi-hop path to be relayed by several VNSs. The single-hop path between two common nodes (the red arrow in Fig. 3) is a single-hop response from the response node to the requesting node when

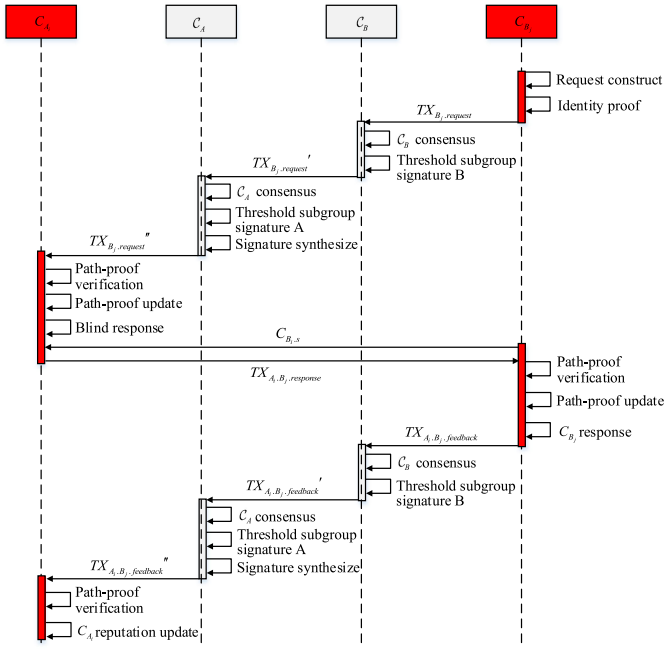

Fig. 4. Cross-chain path signature construction time sequence.

the requesting node issues the value transfer key s to the response node on the basis of the path-proof between the requesting node and the responding node.

To simplify the description, we assume that there are only two privileged subgroups in group $\mathcal{C}$. The cross-chain path signature construction time sequence is shown in Fig. 4. The cross-chain path-proof group signature protocol is denoted as $(t_A, n_A; t_B, n_B; t, n)$. The PPC from both intra-chain and cross-chain is presented below.

Intra-chain PPC. By Definition 3, a recursive path-proof generation formula based on the threshold signature is proposed:

$$\begin{aligned} &u_0 \xrightarrow{TX_1} u_1 : TX_1 = m \| E_{SK_{\mathcal{C}_0}}(m, t_0, n_0), \\ &u_i \xrightarrow{TX_{i+1}} u_{i+1} : TX_{i+1} = m \| E_{SK_{\mathcal{C}_i}}(TX_i, t_i, n_i), i > 0 \end{aligned} \tag{9}$$

(Note: The parameter t, n in Equation (9) is optional and used for threshold signature.)

In Fig. 4, node C_{B_j} constructs request m_{B_j} for another consortium node C_{A_i} in a plaintext manner, and the single-hop path-proof from C_{B_j} to $\mathcal{C}_B$ is constructed as Equation (10) according to Rule 2 and Equation (9). Then, a smart contract transaction $TX_{B_j.\text{request}}$ is generated and deployed in the consortium chain C_B.

$$C_{B_j} \xrightarrow{TX_{B_j.\text{request}}} C_B : TX_{B_j.\text{request}} = m_{B_j} \| E_{SK_{C_{B_j}}}(m_{B_j}) \tag{10}$$

$\mathcal{C}_B$ in C_B is to reach a consensus on transaction $TX_{B_j.\text{request}}$. If Equation (7) is true, i.e., the transaction $TX_{B_j.\text{request}}$ passes the VNL consensus in C_B, then $E_{SK_{\mathcal{C}_B}}(TX_{B_j.\text{request}}, t_B, n_B)$ calculated by Equation (7) is used as the threshold subgroup signature of transaction $TX_{B_j.\text{request}}$ signed by $\mathcal{C}_B$. $E_{SK_{\mathcal{C}_B}}(TX_{B_j.\text{request}}, t_B, n_B)$ and m_{B_j} are written into the smart contract transaction $TX'_{B_j.\text{request}}$ by $\mathcal{C}_B$ according to Equation (11), and the transaction path-proof is updated.

$$\begin{aligned} \mathcal{C}_B &\xrightarrow{TX'_{B_j.\text{request}}} \mathcal{C} : TX'_{B_j.\text{request}} \\ &= m_{B_j} \| E_{SK_{\mathcal{C}_B}}(TX_{B_j.\text{request}}, t_B, n_B) \end{aligned} \tag{11}$$

Cross-chain PPC. In Fig. 4, the path-proof of $TX'_{B_j.\text{request}}$ is verified by $\mathcal{C}_A$ to reach a consensus of $TX'_{B_j.\text{request}}$. If Equation (7) is TRUE, i.e., transaction $TX'_{B_j.\text{request}}$ passes the internal consensus of C_A, then $E_{SK_{\mathcal{C}_A}}(TX'_{B_j.\text{request}}, t_A, n_A)$ calculated by Equation (7) is used as the threshold subgroup signature of transaction $TX'_{B_j.\text{request}}$ signed by $\mathcal{C}_A$. Since the response node $C_{A_i} \in C_A$, each threshold subgroup signature in the path-proof is synthesized by $\mathcal{C}_A$ with the group key.

The construction of the cross-chain path-proof group signature includes three parts: group key generation and sharing, threshold subgroup signature generation, and path-proof update.

Group key generation and sharing. The security prime numbers u, v are selected by the key issuing authority, and $v|(u-1)$ is satisfied. Then, three polynomials $f(x), g_A(x), g_B(x)$ are secretly selected on the finite field Z_v, and the powers in order are $(t-1), (t_A-1), (t_B-1)$. The primitive element α of the finite field Z_v is selected. Then, (u, v, α) and $x_i, y_{A_j}, y_{B_k} \in_R Z_v, i = 1, 2, \ldots, n; j = 1, 2, \ldots, n_A; k = 1, 2, \ldots, n_B$ are displayed. The group private key is randomly generated by the key issuing authority according to Equation (12). Then, the group public key is calculated according to Equation (13). The group private key is distributed by a Shamir-based secret sharing algorithm.

$$SK_c = (f(0) + g_A(0) + g_B(0)) \bmod v \tag{12}$$

$$PK_{\mathcal{C}} = \alpha^{(f(0)+g_A(0)+g_B(0)) \bmod v} \bmod u \tag{13}$$

The group private key fragment $f(x_i)$, $g_A(y_{A_j})$, $g_B(y_{B_k})$ is secretly assigned to $\mathcal{C}_i$ on the basis of the secret sharing algorithm. Then, its public key is calculated and disclosed according to Equation (14).

$$PK_{\mathcal{C}_i} = \alpha^{\lambda_i f(x_i) + \mu_i \sum_X g_X\left(y_{X_{i_j}}\right)} \bmod u \tag{14}$$

Threshold subgroup signature generation. From Equation (5), the number of nodes of subgroup $\mathcal{C}_X$ is $|\mathcal{C}_X| = n_X$, $(n_X > 0, X \in \{A, B\})$ and the threshold number of nodes that pass a certain verification is $t_X \in n_X$. The signed transaction is TX. For each $t_i \in \{t_X\}, \mathcal{K}_i \in Z_u^*$ is secretly and randomly selected. The public key segment r_{X_i} is calculated by Equation (15), the subgroup public key r_X is calculated by $\mathcal{C}_i$ according to Equation (16), and each verification node private key fragment S_{X_i} is calculated by Equation (17), where λ_i, μ_i are the Lagrangian coefficients that are publicly calculated in the Shamir secret sharing algorithm, and $h(x)$ is a safe hash function.

$$r_{X_i} = \alpha^{\mathcal{K}_i} \bmod u, (\mathcal{K}_i \in Z_u^*) \tag{15}$$

$$r_X = \prod_{i=1}^{t_X} r_{X_i} \bmod u \tag{16}$$

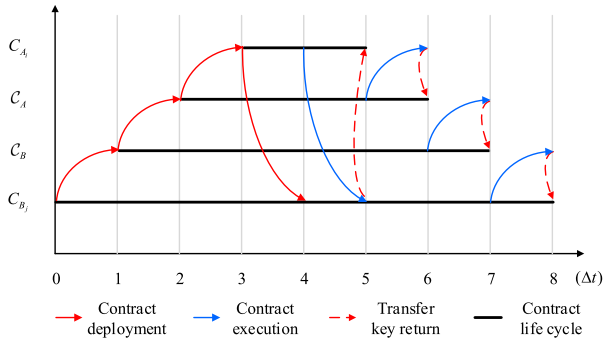

Fig. 5. Value transfer smart contract life cycle.

$$s_{X_i} = \left(f(x_i)\,\lambda_i h(TX) + \sum_X g_X\left(y_{X_{i_j}}\right)\mu_i h(TX) - \mathcal{K}_i r_X \right) \bmod v, X \in \{A, B\} \tag{17}$$

The validity of a single verification node signature s_{X_i} in a subgroup is verified by Equation (18):

$$\alpha^{s_{X_i} r_i^{r_X}} = PK_{C_i}^{h(TX)}, X \in \{A, B\} \tag{18}$$

If Equation (18) is true, the VNS accepts the single verification node signature in the subgroup, and s_X is calculated by Equation (19) when $|s_{X_i}| \geq t_X$. In Fig. 5, the threshold subgroup signature (r_B, s_B) for transaction $TX'_{B_j.\text{request}}$ is calculated by $\mathcal{C}_B$ according to the above-mentioned method. Similarly, since the response node $C_{A_i} \in C_A$, the subgroup synthesis signature $(r_\mathcal{C}, s_\mathcal{C})$ is generated by $\mathcal{C}_A$ according to Equation (20).

$$s_X = (s_{X_1} + s_{X_2} + \cdots + s_{X_i}) \bmod v, X \in \{A, B\} \tag{19}$$

$$\begin{cases} s_\mathcal{C} = (s_{A_1} + s_{A_2} + \cdots + s'_{A_{t_A}} + s_{B_1} + s_{B_2} \\ \qquad + \cdots + s'_{B_{t_B}}) \bmod v, \\ (n_A + n_B) \geq (t'_A + t'_B) \geq t, t'_A \geq t_A, t'_B \geq t_B, \\ r_\mathcal{C} = \prod_X r_X \bmod u. \end{cases} \tag{20}$$

Path-proof update. After the subgroup signatures are synthesized, the group signature of the transaction $TX'_{B_j.\text{request}}$ and m_{B_j} are written into transaction $TX''_{B_j.\text{request}}$ by according to Equation (21). Then, the transaction path-proof is updated.

$$\mathcal{C}_A \xrightarrow{TX''_{B_j.\text{request}}} C_A : TX''_{B_j.\text{request}} = m_{B_j} \| E_{SK_{C_A}}\left(TX'_{B_j.\text{request}}, t_A, n_A; t_B, n_B; t, n\right) \tag{21}$$

The response node C_{A_i} verifies the transaction $TX''_{B_j.\text{request}}$. After it passes, request m_{B_j} is calculated by C_{A_i} using local asset $\xi_{C_{A_i}}$ to generate a blind response $C_{A_i.m_{B_j}.\text{response}}$. Then, the blind response to transaction $TX''_{B_j.\text{request}}$, the updated path-proof, and m_{B_j} are written into transaction $TX_{A_i.B_j.\text{response}}$ according to Equation (22).

$$C_{A_i} \xrightarrow{TX_{A_i.B_j.\text{response}}} C_{B_j} : TX_{A_i.B_j.\text{response}} = m_{B_j} \left\| E_{SK_{C_{A_i}}}\left(TX''_{B_j.\text{request}}\right) \right\| C_{A_i.m_{B_j}.\text{response}} \tag{22}$$

After receiving transaction $TX_{A_i.B_j.\text{response}}$, C_{B_j} verifies its path-proof. If it passes, C_{B_j} sends the value transfer key $C_{B_j.s}$ to $TX_{A_i.B_j.\text{response}}$ to trigger the transaction. Then, the transaction $TX''_{A_i.B_j.\text{feedback}}$ containing feedback from C_{B_j} is generated and broadcasted similar to that in Equations (10), (11), (21), so that the institution can motivate the responder.

C. Value Transfer Mechanism (VTM)

Based on a previous study on the reputation incentive mechanism [27], the institution sets (i) an integral mechanism for constraining the value transfer of the smart contract within the consortium chain and (ii) a heterogeneous integral value exchange function between different chains. Equal value transformation of heterogeneous assets is achieved by calling this function in smart contracts. In the consortium chain, the node public-private key is used as the identifier. When a node enters the system for the first time, the institution allocates a certain number of points for starting. In each transaction, the requesting node needs to deposit a certain number of points in the smart contract and set the transfer condition as the value incentive. When the transfer condition is triggered, the point is transferred irreversibly; accordingly, the request node reduces the deposited quantity integral and the response node obtains the equivalence integral. Moreover, as nodes with a score of 0 are unable to provide incentives, they risk being starved to death without being responded to for long periods of time. The manner in which value is transferred cross-chain is described below.

Cross-chain value transfer is realized by deploying smart contracts. One deployment of smart contracts is called a stage of inter-chain value transfer. If cross-chain communication smart contract deployment is divided into $k(k > 1)$ stages, correspondingly, the execution process also has k stages. By Definition 2, after the previous stage is deployed or executed at least Δt times, the system reaches a stable state before deploying or executing the next stage. Otherwise, there will be cases where the smart contract from the previous stage is not deployed or not executed, and the latter stage of the smart contract cannot be executed, causing some node losses. Therefore, a rational node will deploy and execute the smart contract with Δt as the minimum time interval:

- If the time of deploying the first stage of the cross-chain communication smart contract is $t_{\text{timestamp}}$, accordingly, the time of smart contract deployment in each stage is $\{t_{\text{timestamp}}, t_{\text{timestamp}} + \Delta t, \ldots, t_{\text{timestamp}} + (k-1)\Delta t\}$.
- The smart contract execution order triggered by the value transfer key is opposite to the deployment order, and the execution starting time of the corresponding smart contracts is $t_{\text{timestamp}} + \{(2k-1)\Delta t, (2k-2)\Delta t, \ldots, k\Delta t\}$.
- The smart contract life cycle with hash lock is $\{2k\Delta t, (2k-1)\Delta t, \ldots, 2\Delta t\}$.
- In systems that require timely communication, such as IoT smart devices, the effective deadline for smart contracts in each stage is $t_{\text{timestamp}} + \{2\,k\Delta t, (2k-1)\Delta t, \ldots, (k+1)\Delta t\}$.

In Fig. 4, the cross-chain communication value transfer smart contract deployment is divided into four stages (exclud ing feedback incentives). The smart contract life cycle of each stage is

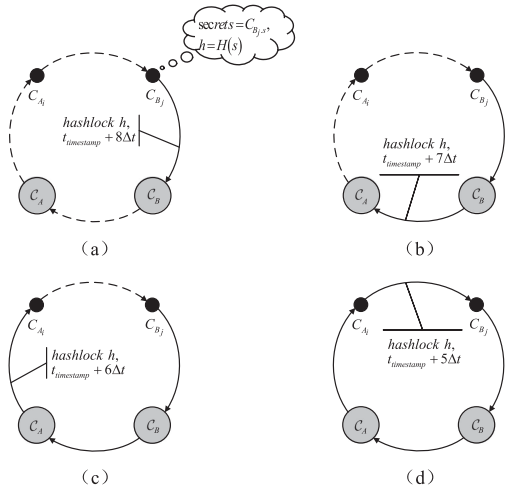

Fig. 6. Value transfer smart contract deployment time sequence.

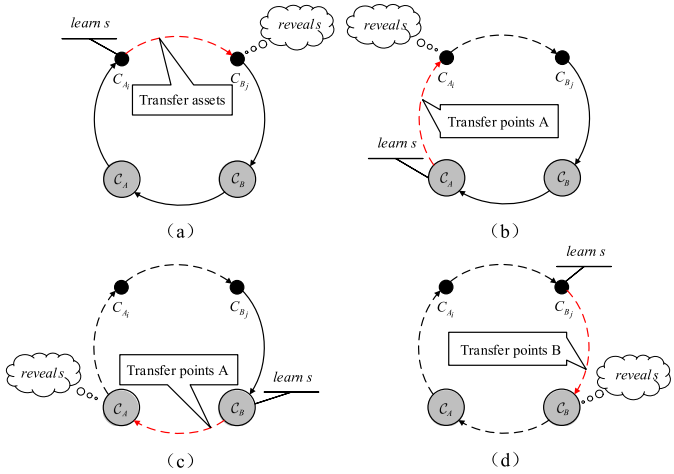

Fig. 7. Value transfer smart contract trigger time sequence. (*Note:* Solid arrow in the figure indicates that the smart contract has been deployed, black dotted arrow indicates that the smart contract has been executed, and red dotted arrow indicates that the smart contract is being executed.)

shown in Fig. 5, where the horizontal axis represents the timeline in Δt and the vertical axis represents the nodes participating in contract deployment and execution. The red arrow indicates the smart contract deployment negotiated by the head and tail nodes of the arrow, and the arrow direction indicates the transfer of the promised value in the smart contract. The blue arrow indicates that the corresponding node of the arrow header sends the value transfer key $C_{B_j.s}$ to trigger the execution of the contract at the corresponding time of the arrow tail. The arrow direction indicates the direction in which the value in the smart contract is transferred. After the execution of the smart contract, the value transfer key is sent to the node indicated by the red dotted arrow in the form of a return value to trigger the subsequent smart contract execution. The black bold line represents the smart contract life cycle with the corresponding node on the vertical axis as the originating node.

A rational node expects the contract to be executed without losing its own interests; hence, it tends to trigger the contract as soon as possible after obtaining the value transfer key and realize the value transfer. Figs. 6 and 7 show a more detailed description of the smart contract deployment, triggering, and execution shown in Fig. 5.

The major process of VTM is as follows:

Step 1: As shown in Fig. 6(a), request node C_{B_j} creates value transfer key $C_{B_j.s}$, selects one-way anti-collision hash function $H(\cdot)$, and calculates $h = H(C_{B_j.s})$. C_{B_j} negotiates with $\mathcal{C}_B$ to generate a smart contract with hash lock h and time lock $t_{\text{timestamp}} + 8\Delta t$, and it deploys the contract to achieve C_{B_j} to $\mathcal{C}_B$ point transfer depositing. If $\mathcal{C}_B$ sends value transfer key $C_{B_j.s}$ to the smart contract before time $t_{\text{timestamp}} + 8\Delta t$ and $h = H(C_{B_j.s})$ is true, then the points deposited in the smart contract will be irrevocably transferred from C_{B_j} to $\mathcal{C}_B$; if $\mathcal{C}_B$ cannot reveal the secret before time $t_{\text{timestamp}} + 8\Delta t$, a refund transaction is executed by C_{B_j}, and the points deposited in the smart contract will be refunded to C_{B_j}.

Step 2: As shown in Fig. 6(b), after $\mathcal{C}_B$ confirms that the smart contract generated in step 1 is deployed stably, $\mathcal{C}_B$ negotiates with $\mathcal{C}_A$ to generate a smart contract with hash lock h and time lock $t_{\text{timestamp}} + 7\Delta t$, as well as the same value as the contract deployed by Step 1 calculated by the preset heterogeneous value exchange function between the consortium chains. Then, $\mathcal{C}_B$ deploys the smart contract deposited point of C_A to achieve C_{B_j} to $\mathcal{C}_B$ point transfer depositing. The execution logic of the smart contract in Step 2, 3, and 4 is similar to that in Step 1 and will not be described again.

Step 3: As shown in Fig. 6(c), after $\mathcal{C}_A$ confirms that the smart contract generated in Step 2 is deployed stably, $\mathcal{C}_A$ negotiates with C_{A_i} to generate a smart contract with hash lock h and time lock $t_{\text{timestamp}} + 6\Delta t$, as well as the same value as the contract deployed by Step 2. Then, it deploys the smart contract to achieve $\mathcal{C}_A$ to C_{A_i} point transfer depositing.

Step 4: As shown in Fig. 6(d), after C_{A_i} confirms that the smart contract generated in Step 3 is deployed stably, C_{A_i} negotiates with C_{B_j} to generate a smart contract with hash lock h and time lock $t_{\text{timestamp}} + 5\Delta t$ as well as the same value as the contract deployed by Step 2. Then, it deploys the smart contract to achieve C_{A_i} to C_{B_j} point transfer depositing.

Step 5: As shown in Fig. 7(a), after C_{B_j} confirms that the smart contract generated in Step 4 is deployed stably, C_{B_j} reveals the value transfer key s to the contract within the validity period of the contract. Then, the contract is triggered to execute, and C_{B_j} obtains the assets deposited by C_{A_i} in the contract. After the contract is executed, s is returned to the smart contract deployer C_{A_i}.

Step 6: As shown in Fig. 7(b), similarly to Step 5, after C_{A_i} learns s, C_{A_i} reveals s to the contract deployed in Step 3 within the validity period of the contract. Then, the contract is triggered to execute, and C_{A_i} obtains the assets deposited by $\mathcal{C}_A$ in the contract. After the contract is executed, s is returned to the smart contract deployer $\mathcal{C}_A$.

Step 7: As shown in Fig. 7(c), after $\mathcal{C}_A$ learns s, $\mathcal{C}_A$ reveals s to the contract deployed in Step 2 within the validity period of the contract. Then, the contract is triggered to execute, and

$\mathcal{C}_A$ obtains the assets deposited by $\mathcal{C}_B$ in the contract. After the contract is executed, s is returned to the smart contract deployer $\mathcal{C}_B$.

Step 8: As shown in Fig. 7(d), after $\mathcal{C}_B$ learns s, $\mathcal{C}_B$ reveals s to the contract deployed in Step 1 within the validity period of the contract. Then, the contract is triggered to execute, and $\mathcal{C}_B$ obtains the assets deposited by C_{B_j} in the contract.

Steps 1–8 realize value transfer between different consortium chains. The smart contract value transfer algorithm is shown in Algorithm 1. The algorithm is developed according to the following appointment. The value transfer smart contract only accepts and verifies the value transfer key provided by the specified transaction partner, and it only returns the value transfer key to the node that provides a legal path-proof. Further, the refund function only accepts and verifies the call from the asset deployer of the contract.

Algorithm 1: Smart Contract Value Transfer Algorithm.

Input: transaction responder (i.e., asset deployer) $responder$, transaction sponsor $requester$, deployed asset ξ, path-proof triple (m, p, σ) of $responder$, value transfer key s, current time t_{current}.

Output: If the contract is triggered within the validity period, the contract executes so that the assets deployed in the contract are transferred to $requester$ and value trigger key s is returned to $responder$. Otherwise, $responder$ calls the refund transaction to return the asset deployed in the smart contract to $responder$.

```
Contract Asset _ transfer {
bool locked; address responder, requester; asset ξ;
unti t_timelock, h_hashlock, unit t_timestamp;
    Function Deploy(address responder', requester';
     asset ξ';
     unti t'_timelock, h'_hashlock){
     responder = responder'; requester = requester';
     ξ = ξ';
     t_timelock = t'_timelock; h_hashlock = h'_hashlock; locked = true;
     }
     Function Asset_transfer(unit s; path p; sig σ){
     Require(s.sender = requester);
     If(t_current ≤ t_timelock) ∩ (h_hashlock == h(s))∩
     isPath(p) ∩ isSig(m, p, σ) ∩(locked == true)
      {locked == false;
      Transfer asset to requester;
      Return s to responder;
      }
     Else
      Halt;
  }
  Function Refund() {
    Require(msg.sender == responder) ∩ (t_current >
     t_timelock);
    If(locked == true)
     Transfer asset to responder;
     Return;
  }
}
```

IV. Analysis and Verfication

A. Performance Analysis

Security analysis. Suppose that there are actually t nodes in group $\mathcal{C}$ that sign transaction TX, and there are at least t_A nodes from $\mathcal{C}_A$ and t_B nodes from $\mathcal{C}_B$. Then, Equation (23) is established.

$$\begin{aligned} s_{\mathcal{C}} &= h(TX)\left(\sum_{i=1}^{t} f(x_i)\lambda_i + \sum_{j=1}^{t_A} g_A\left(y_{A_j}\right)\mu_i \right. \\ &\quad \left. + \sum_{k=1}^{t_B} g_B\left(y_{B_k}\right)\mu_i\right) r_{\mathcal{C}} \sum_{i=1}^{t} \mathcal{K}_i \\ &= h(TX)\left(f(0) + g_A(0) + g_B(0)\right) - r_{\mathcal{C}} \sum_{i=1}^{t} \mathcal{K}_i \end{aligned} \tag{23}$$

Therefore, the verification equation, i.e., Equation (24) is established.

$$\alpha^{s_{\mathcal{C}}} r_{\mathcal{C}}^{r_{\mathcal{C}}} = PK_{\mathcal{C}}^{h(TX)} \tag{24}$$

It can be seen from Equations (23) and (24) that nodes not in group $\mathcal{C}$ cannot participate in or interfere with the above- mentioned verification process, the forged non-cooperative cross-chain communication path-proof will not be verified, and the system will ignore the corresponding transaction. If the number of nodes participating in signature in C_{X_i} is less than t, it is possible to recover $g_X(0), X \in \{A, B\}$, but the component $f(0)$ cannot be recovered; hence, the group private key cannot be recovered and verified. If the number of nodes participating in signature in $\mathcal{C}$ is greater than or equal to t and the number of nodes participating in verification in subgroup $\mathcal{C}_X$ is less than t_X, it is possible to recover $f(0)$, but the component $g_X(0)$ cannot be recovered; hence, the group private key still cannot be recovered and verified. Therefore, the threshold group signature mechanism based on privileged subgroups can realize the identity validity proof of communication between consortium chains, thus improves the system security.

Scalability analysis. The cross-chain protocol TCCM based on PPC $(t_A, n_A; t_B, n_B; t, n)$ can be easily extended to multi- privileged subgroup path-proof group signature protocol $(t_A, n_A; t_B, n_B; \cdots; t_X, n_X; t, n)$ with more consortium chains participating. The extended method is as follows. In the protocol establishment stage, $|\{A, B, \ldots, X\}| + 1$ polynomials $f(x), g_A(x), g_B(x), \ldots, g_X(x)$ are selected. The extended group private key and public key calculation equations can be derived from Equations (12) and (13) as follows:

$$SK_c' = \left(f(0) + \sum_{i=A}^{X} g_i(0)\right) \bmod v, i \in \{A, B, \ldots, X\} \tag{25}$$

$$PK_c' = \alpha^{\left(f(0)+\sum_{i=A}^{X} g_i(0)\right) \bmod v} \bmod u \tag{26}$$

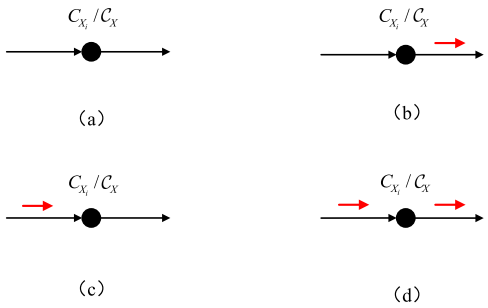

Fig. 8. Four possible scenarios for value transfer.

In Equations (25) and (26), the node in each privileged subgroup holds the component $f(x_i)$ and the corresponding component $g_X(y_{X_j})$. The verification process is the same as that described in Section III.

The directed connected path signature $\sigma = \text{sig}(\cdots, \text{sig}(m, u_0), \ldots, u_k)$ from request node u_0 to response node u_k includes the single node signature of u_0, u_k and some relay subgroup threshold signatures on the path. In this mechanism, the reliability of node identity is guaranteed by the consortium chain VNL subgroup consensus where the single node is located. Security of the threshold group signature mechanism provides a security guarantee for cross-chain communication while it limits the communication to only the cooperated consortium chains. The system has good scalability.

Game analysis of value transfer. Fig. 8 shows four possible scenarios for value transfer at node C_{X_i} (or $\mathcal{C}_X$) by triggering the smart contract from the node cooperative game perspective. The black arrow indicates that the value transfer smart contract has been deployed and the red arrow indicates that when the interaction ends, only the corresponding smart contract is triggered and its assets are transferred.

In Fig. 8(a), smart contracts corresponding to the inbound and outbound edges of the node are both not triggered, indicating that no value is transferred through the node and the balance of the node has not changed. However, it conflicts with the willingness of rational nodes to participate and reach an interaction.

In Fig. 8(b), only the smart contract corresponding to the outbound edge of the node is triggered, and the contract corresponding to the inbound edge of the node is not triggered. Thus, the node suffers losses. Therefore, excluding the complex situation in which the node is latent in the system and deliberately engages in other types of attacks, rational nodes will not choose this strategy.

In Fig. 8(c), only the smart contract corresponding to the inbound edge of the node is triggered, and the contract corresponding to the edge of the node is not triggered. Thus, the assets of the node increase. Hence, if there is no constraint, the node will select this approach to maximize its own interests.

In Fig. 8(d), smart contracts corresponding to the inbound and outbound edges of the node are both triggered, and the value of the node participating in value transfer is conserved. This is the normal situation that rational nodes would like to achieve. From the above-mentioned analysis, the best way for rational nodes to perform selection is shown in Fig. 8(c); however, the reasonable value transfer strategy between different medical consortium chains is shown in Fig. 8(d).

In VTM proposed in this paper, the nodes participating in the cross-chain value transfer include the requesting node, the responding node, and the relay node. In Fig. 6, when the requesting node C_{B_j} observes that its inbound smart contract is deployed stably by C_{A_i}, C_{B_j} sends the value transfer key $C_{B_j.s}$ to the contract to trigger value transfer. When C_{A_i} observes that its inbound smart contract is deployed stably by $\mathcal{C}_A$, it would deploy its outbound smart contract, and so on.

The above-mentioned mechanism ensures that the value transfer smart contract between nodes is sequentially deployed within time $t_{\text{timestamp}} + \{k\Delta t, (k-1)\Delta t, \ldots, \Delta t\}$ (here, $k = 4$). As shown in Fig. 7, the expiry time of value transfer smart contracts deployed counterclockwise starting from C_{B_j} between adjacent communication nodes is $t_{\text{timestamp}} + \{(k+1)\Delta t, (k+2)\Delta t, \ldots, 2k\Delta t\}$.

The value transfer game process of nodes on the communication link from C_{B_j} to C_{A_i} is analyzed as follows. The initial state of C_{B_j} is shown in Fig. 8(a). As a rational transaction requesting node, C_{B_j} wishes to achieve cross- chain interaction. Therefore, C_{B_j} will send the value transfer key $C_{B_j.s}$ to its inbound smart contract before its expiry time to trigger value transfer. C_{B_j} will temporarily enter the state in Fig. 8(c), and the value transfer key will be returned to C_{A_i} after the smart contract is executed. Then, C_{A_i} will temporarily enter the state shown in Fig. 8(b). As C_{A_i} is rational, once it obtains the value transfer key, it will send the key to its incoming edge smart contract during its lifetime and convert its state to that shown in Fig. 8(d). After the smart contract is executed, the value transfer key is returned to $\mathcal{C}_A$. Similarly, the state of $\mathcal{C}_A$, $\mathcal{C}_B$ is finally converted from that shown in Fig. 8(b) to that shown in Fig. 8(d), and the state of C_{B_j} is converted from that shown in Fig. 8(c) to that shown in Fig. 8(d) by triggering the incoming edge smart contract during the validity period.

It can be seen from the above-mentioned analysis that VTM proposed in this paper can ensure value transfer between different medical consortium chains by the deployment of smart contracts. If all the parties involved comply with the mechanism proposed in this paper, cross-chain value interaction can be achieved by triggering the smart contract. If there is a node that violates the mechanism, only that node will suffer.

B. Experimental Deployment

To test the feasibility and performance of TCCM based on PPC proposed in this paper, we built an Ethereum simulation test environment consisting of 230 virtual verification nodes on five servers. The experimental platform is as follows: CPU, Intel Xeon E5; memory size, 64 GB; operating system, Ubuntu 64-bit.

The consensus process consists of four parts: node key generation, key reconstruction calculation, consensus signature, and consensus signature verification. The node key generation is obtained by pre-calculation between nodes; hence, it does not add to the network delay. The actual network delay is mainly affected by key reconstruction calculation, consensus signature, and consensus signature verification. Therefore, the network

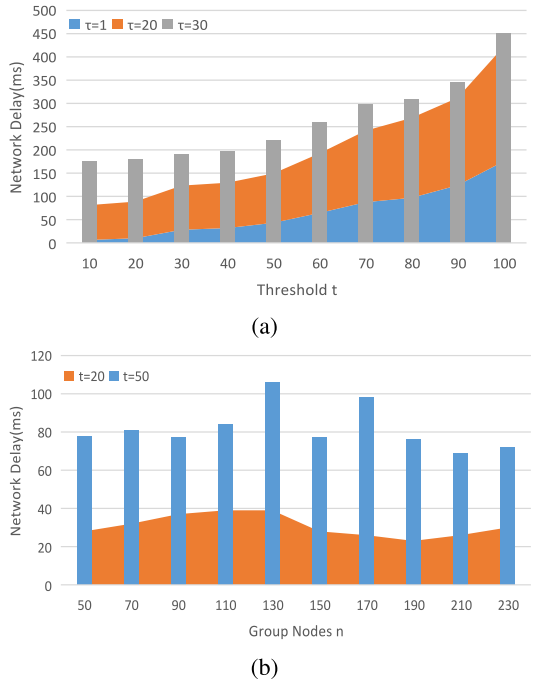

Fig. 9. Single-chain performance test.

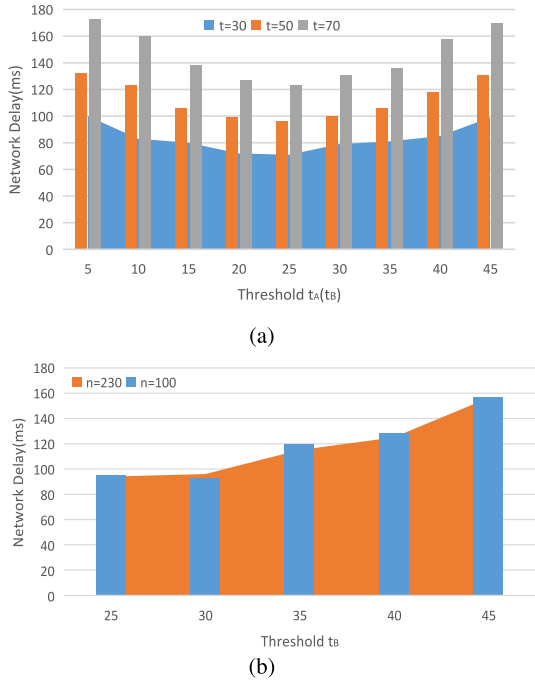

Fig. 10. Cross-chain performance test.

latency calculation method is to sum the time costs of these three parts.

Single-chain performance. In a single-chain environment, the number of verification nodes $n = 230$, and the system constructs transactions at intervals of 10 ms. When threshold t takes different values in the interval [10,100] with a step size of 10, and the number of transactions in a single consensus takes $\tau = 1, 20, 30$, the average network delay of individual transaction varies with the threshold as shown in Fig. 9(a). It can be seen that (i) the network delay increases significantly when $t \in (50, 100]$ and (ii) increasing the number of transactions in a single consensus is beneficial for improving the system throughput. However, it will result in an increase in the average network delay for individual transactions. Therefore, by taking $t = 20$ and $t = 50$ separately, the network delay variation is tested when the total number of verification nodes n varies in the interval [50,230] with a step size of 20. The results are shown in Fig. 9(b). It can be seen that the network delay fluctuates with n but is basically stable. Thus, the network delay is not affected by n significantly but it is mainly affected by t.

Cross-chain performance. We constructed two consortium chains with $n = 230, n_A = n_B = n/2, t_A = t_B = t/2$ to compare the performance of the chains. When $t = 30$, $t = 50$, $t = 70$, and threshold t_B takes different values in the interval [5,45] with a step size of 5, the actual cross-chain network delay varies with the threshold as shown in Fig. 10(a). It can be seen that when the group threshold t is fixed, as the subgroup threshold value changes, inter-chain network delay occurs regularly. Multiple tests show that the network delay is smaller and the performance is better when $t_X \to t/m$ ($X \in \{A, B, \ldots\}, m > 1, m$ is the number of cooperative consortium chains).

The test result of inter-chain communication delay when the VNLs of all consortium chains have the same scale and the privileged subgroups have the same rights is presented above. From the result in Section IV of the single-chain performance test, the network delay is not significantly affected by VNL number n but it is mainly affected by the threshold t. Thus, it can be concluded that when the scales of the consortium VNL are different, and the group threshold $t_X \to t/m$ ($X \in \{A, B, \ldots\}, m > 1$), the inter-chain communication still achieves better performance.

Consider a situation of unequal rights between the partners of the consortium chains in an actual application scenario. Hence, t_A is fixed while t_B varies, and the system is tested when $t_A + t_B > t$. Reset the parameters of the above-mentioned consortium chains :$n = 100, t = 50, n_A = n_B = 50, t_A = 25$. The network delay is tested when t_B varies in the interval [25,45] with a step size of 5; then, the experiment is repeated when $n = 230$. The results are shown in Fig. 10(b). In this scenario, as t_B increases, the cross-chain network latency fluctuates to some extent, but the general trend is to increase. Therefore, it can be concluded that on the basis of the equal rights between the consortium chain partners, privilege escalation of one chain will increase the network delay between the chains. However, the increase in the number of verification nodes has no obvious effect on the network delay.

Smart contract deployment and execution. In the above-mentioned simulation test environment, lightweight smart contracts are constructed according to Algorithm 1, and a complete cross-chain interaction is realized by deploying 8-stage smart contracts with the hash lock function SHA 256. According to the communication performance test results presented above, the upper limit of the response time Δt between nodes is set to

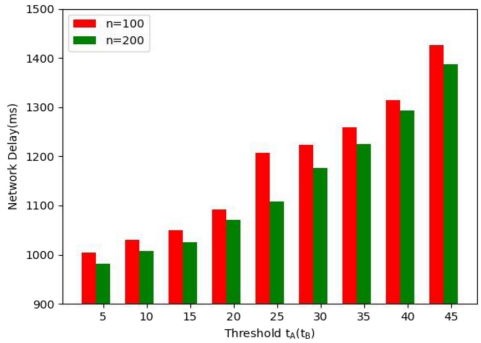

Fig. 11. 4-Stage smart contracts deployment delay.

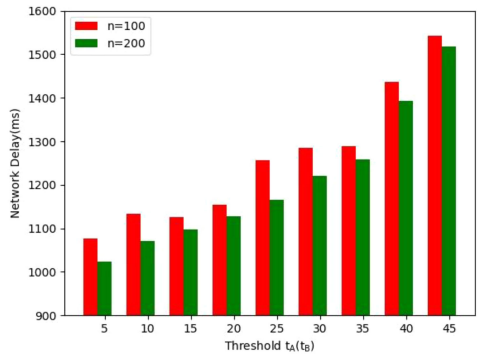

Fig. 12. 4-Stage smart contracts execution delay.

160 ms. For threshold values $t = t_A + t_B, t_A = t_B$, the number of verification nodes takes different values of $n = 100$ and $n = 200$, which are tested separately, the result of 4-stage smart contracts deployment delay is shown in Fig. 11.

The above-mentioned 4-stage smart contracts deployment delay is mainly due to three factors: the intra-chain consensus, the cross-chain consensus, and the upper limit of the response time Δt of the nodes to the contract deployment. As can be seen from Fig. 11, under the assumption that the upper response interval Δt of the nodes to the contract deployment is a constant, (i) the network delay increases with the threshold value when the number of verification nodes is $n = 100$ and $n = 200$, and (ii) if the threshold value of the consensus mechanism is a constant, increasing the number of verification nodes can reduce the network delay in a certain range.

Under the two test cases $n = 100$ and $n = 200$, the execution delay of the 4-stage contracts are tested separately. After the smart contracts are deployed in each stage, the value transfer key is revealed by the node from which the request originates to the smart contract deployed in the fourth stage. Then, the remaining stages of the contracts are triggered in turn. The 4-stage smart contracts execution delay is shown in Fig. 12. It can be seen from the graph that with the increase in the consensus threshold value, the execution process network delay of the 4-stage smart contracts presents a change similar to its deployment process. However, there is a slight increase in the execution process network delay of the 4-stage smart contracts in comparison with its deployment process network delay. This is mainly because in addition to the delay caused by the consensus within and between the chains, the 4-stage contracts execution delay is also affected by the calculation of the function SHA 256 and the actual response delay of nodes on the communication path.

In the proposed cross-chain interaction mechanism, a complete cross-chain interaction network delay includes the above-mentioned 4-stage smart contracts deployment delay and its execution delay, which is mainly affected by the consensus threshold value and the upper limit of the response time Δt of nodes to the contract deployment and execution. Nevertheless, under the existing terminal computing capacity, the hash operation for value transfer locking brings a shorter delay. Therefore, the proposed scheme facilitates the use of the unidirectivity of the high-security hash algorithm to guarantee reliability of the value transfer between multi-stage smart contracts. In addition, a complete cross-chain interaction takes place within seconds in the test environment, which basically meets the response requirements between medical consortium chains.

V. Conclusion

Consortium chains have considerable potential for medical applications. To optimize the collaboration between heterogeneous medical consortium chains, we improved the dynamic autonomous interaction cross-chain from three aspects: TCCM, PPC, and VTM. Based on the consensus mechanism VNL within consortium chain, consensus between multiple cooperative medical consortium chains was modeled as a threshold digital signature process with multiple privileged subgroups. By simplifying the P2P communication topology, the PPC rules and the recursive path-proof generation formula based on the threshold signature were proposed to construct the cross-chain node identity credibility proof. We analyzed the deployment and execution time sequence of the smart contract from the perspective of the node cooperative game, determined the best way for value transfer of rational nodes, and proposed the life cycle of the smart contract with hash locking. Experimental results showed that the proposed scheme can achieve autonomous dynamic cross-chain interaction within a tolerable network delay on the basis of satisfying data privacy and security requirements. However, our experiment was carried out under the condition that the scale of the consortium chain was limited, the actual performance and the cost of building a system is unclear. Its promotion requires medical institutions, governments, and other relevant departments to invest heavily in storage and management standards, policy formulation, and infrastructure construction of medical data.

References

[1] L. Sweeney, "Simple demographics often identify people uniquely," *Health* (San Francisco), vol. 671, pp. 1–34, 2000.

[2] D. Munro, "Data breaches in healthcare totaled over 112 million records in 2015. [Online]. Available: https://www.forbes.com/sites/danmunro/2015/12/31/data

[3] K. Singh and L. Batten, "Aggregating privatized medical data for secure querying applications," *Future Gener. Comput. Syst.*, vol. 7, no. 72, pp. 250–263, 2017.

[4] F. Jabeen *et al.*, "Enhanced architecture for privacy preserving data integration in a medical research environment," *IEEE Access*, vol. 5, pp. 13308–13326, 2017.

[5] H. A. Al Hamid, S. M. M. Rahman, M. S. Hossain, A. Almogren, and A. Alamri, "A security model for preserving the privacy of medical big data in a healthcare cloud using a fog computing facility with pairing-based cryptography," *IEEE Access*, vol. 5, pp. 22313–22328, 2017.
[6] H. Arshad and A. Rasoolzadegan, "Design of a secure authentication and key agreement scheme preserving user privacy usable in telecare medicine information systems," *J. Med. Syst.*, vol. 40, no. 11, p. 237, 2016.
[7] K. Abouelmehdi, A. Beni-Hessane, and H. Khaloufi, "Big healthcare data: Preserving security and privacy," *J. Big Data*, vol. 5, no. 1, pp. 1–5, 2018.
[8] G. Zyskind and O. Nathan, "Decentralizing privacy: Using blockchain to protect personal data," in *Proc. IEEE Secur. Privacy Workshops*, 2015, pp. 180–184.
[9] K. WÃijst and A. Gervais, "Do you need a blockchain?" in *Proc. Crypto Valley Conf. Blockchain Technol.*, 2018, pp. 45–54.
[10] E. Zaghloul, T. Li, and J.Ren, "Security and privacy of electronic health records: Decentralized and hierarchical data sharing using smart contracts," in *Proc. Int. Conf. Comput., Netw. Commun.*, 2019, pp. 375–379.
[11] G. G. Dagher, J. Mohler, M. Milojkovic, and P. B. Marella, "Ancile: Privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology," *Sustain. Cities Soc.*, vol. 39, pp. 283–297, 2018.
[12] R. Guo, H. Shi, Q. Zhao, and D. Zheng, "Secure attribute-based signature scheme with multiple authorities for blockchain in electronic health records systems," *IEEE Access*, vol. 6, pp. 11676–11686, 2018.
[13] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," *Proc. IEEE Int. Congr. Big Data*, 2017, pp. 557–564.
[14] H. Wang, Z. Zheng, S. Xie, and H. N. Dai, "Blockchain challenges and opportunities: A survey," *Int. J. Web Grid Services*, vol. 14, no. 4, pp. 352–375, 2018.
[15] D. Bender and K. Sartipi, "HL7 FHIR: An agile and restful approach to healthcare information exchange," in *Proc. 26th IEEE Int. Symp. Comput.-Based Med. Syst.*, 2013, pp. 326–331.
[16] P. Zhang, J. White, D. C. Schmidt, G. Lenz, and S. T. Rosenbloom, "FHIRChain: Applying blockchain to securely and scalably share clinical data," *Comput. Structural Biotechnol. J.*, vol. 16, pp. 267–278, 2018.
[17] PhUSE Emerging Trends & Technology, "How blockchain can transform the pharmaceutical and healthcare industries," 2018. [Online]. Available: https://www.phuse.eu/documents/working-groups/deliverables/phuse-blockchain-white-paper-version-10-final-18719.pdf
[18] M. Li, L. Xia, and O. Seneviratne, "Leveraging standards based ontological concepts in distributed ledgers: A healthcare smart contract example," in *Proc. IEEE Int. Conf. Decentralized Appl. Infrastructures*, 2019, pp. 152–157.
[19] T. T. Kuo, H. E. Kim, and L. Ohno-Machado, "Blockchain distributed ledger technologies for biomedical and health care applications," *J. Amer. Med. Inform. Assoc.*, vol. 24, no. 6, pp. 1211–1220, 2017.
[20] J. Poon and T. Dryja, "The bitcoin lightning network: Scalable off-chain instant payments," 2016. [Online]. Available: https://lightning. network/lightning-network-paper.Pdf
[21] D. Piatkivskyi, S. Axelsson, and M. Nowostawski, "Digital forensic implications of collusion attacks on the lightning network," in *Proc. IFIP Int. Conf. Digit. Forensics*, 2017, pp. 133–147.
[22] BlockStream 2014. [Online]. Available: https://blockstream.com/
[23] A. Back *et al.*, "Enabling blockchain innovations with pegged sidechains," 2014. [Online]. Available: http: //www. opensciencereview.com/papers/123/enablingblockchain-innovations-with -pegged-sidechain-s
[24] V. Buterin, "Ethereum sharding faq," 2018. [Online]. Available: https: //git hub.com/ethereum/wiki/wiki/Sharding-FAQs
[25] V. Buterin, "A next-generation smart contract and decentralized application platform," 2014. [Online]. Available: https://cryptorating.eu/whitepapers/Ethereum/Ethereum_white_paper.pdf
[26] Y. Zhang, S. Kasahara, Y. Shen, X. Jiang, and J. Wan, "Smart contract-based access control for the Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1594–1605, Apr. 2019.
[27] R. Qiao, S. Zhu, Q. Wang, and J. Qin, "Optimization of dynamic data traceability mechanism in Internet of Things based on consortium blockchain," *Int. J. Distrib. Sensor Netw.*, vol. 14, no. 12, pp. 1–15, 2018.