DECS: A Decentralized and Efficient Cross-Chain Scheme in IoT System

Ying Gao(✉), Peihao Zhang, Qiaofeng Pan, and Xianfeng Qiu

School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, People's Republic of China
gaoying@scut.edu.cn

Abstract. The rapid development of blockchain technology has demonstrated great potential to revolutionize various application domains, especially in the context of the Internet of Things (IoT). However, the sheer number and diversity of IoT devices pose significant challenges to the scalability and security of blockchain systems. To address these issues, cross-chain technology has emerged as a promising solution. Nevertheless, existing cross-chain solutions suffer from high centralization and inefficiency. Our approach involves constructing a multi-chain network capable of connecting different types of IoT devices, along with designing a collaborative cross-chain mechanism engaging multiple participating nodes. This collective participation diminishes the centralization inherent in the cross-chain process. Specifically, we propose a data verification method based on BLS signature. It aggregates cross-chain data signatures across multiple chains, which leads to a reduced storage burden on the blockchain. Furthermore, we introduce a reputation update algorithm that leverages network latency and cross-chain operation metrics to automatically update node reputation scores via smart contracts. Experimental results demonstrate that our solution achieves better decentralization and efficiency.

Keywords: blockchain · IoT · cross-chain · BLS signature

1 Introduction

Blockchain is gradually extending from small-scale applications to multiple fields, showing a bright development prospect. It has been applied to finance [1], food traceability [1], smart home [10], IoT, and other industries. In particular, IoT has a very good application prospect with blockchain due to its own decentralized features [5]. However, another feature of IoT is the large number and diversity of devices. With a large number of devices connected to the blockchain system, higher requirements are placed on the performance of the blockchain system. Unlike centralized systems, blockchain requires all nodes to reach a consensus

This work is supported by the Guangzhou Science and Technology Program key projects (202103010005).

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2024
Published by Springer Nature Switzerland AG 2024. All Rights Reserved
V. C. M. Leung et al. (Eds.): Qshine 2023, LNICST 573, pp. 128–142, 2024.
https://doi.org/10.1007/978-3-031-65126-7_13

during the transaction process [15]. So it leads to a significant gap in blockchain performance compared to centralized systems.

Cross-chain technologies are seen as an effective way to address blockchain scalability and performance. It can join different devices to separate blockchain networks and organize multiple blockchains through cross-chain technologies. The performance of blockchains can be effectively improved. Currently, the mainstream cross-chain technologies include sidechains/relays [2], notary schemes [9], and hash-locking [6]. However, all these technologies have shortcomings. Among them, hash-locking and sidechains/relays are mainly used for asset transfer rather than information interaction. This poses the problem that they focus more on security when crossing chains. And reduce the performance requirements. The notary schemes, on the other hand, suffer from the problem of centralization. In general, the notary mechanism is only responsible by fixed nodes in the cross-chain. The trustworthiness of the cross-chain data is completely guaranteed by the node's own credit. If the node carries malicious intent or it receives an attack, the security of the entire cross-chain process cannot be guaranteed.

In this paper, We propose an decentralized and efficient cross-chain scheme (DECS). First, We construct a multi-chain architecture consisting of multiple local-chains and a global-chain. And we add IoT devices to the local-chain network. Mutually independent local-chains can perform block transactions in parallel. It improves the blockchain and performance. Meanwhile, we propose a scheme in which multiple nodes jointly participate in cross-chain data verification. We design a calculation method for node reputation. It is calculated based on the network latency and invocation frequency of the nodes so that the selection of each node is as average as possible. This reduces the degree of centralization in cross-chain and effectively addresses the shortcomings of the notary schemes. During the cross-chain process, multiple nodes with the highest reputation are selected. They are responsible for verifying the data by using BLS signatures. Our major contributions in this article are summarized as the following aspects:

- We propose an Decentralized and Efficient Cross-chain Scheme in IoT systems. We adopt a multi-chain architecture to adapt the diversity of types of IoT devices and enhance the scalability of the blockchain.
- We design a cross-chain scheme based on BLS signatures and implement a smart contract that automatically updates node reputation. The scheme reduces the degree of centralization of the cross-chain process while guaranteeing the accuracy of cross-chain data.
- We implement and evaluate our scheme on the HyperLedger Fabric, and the results shows that our scheme can effectively improve system performance, and the selection of nodes is uniform and fair in the cross-chain process.

The rest of this paper is structured as follows. In Sect. 2, we introduce the related work of blockchain. In Sect. 3, we described the system architecture and cross-chain interaction mechanisms. And we introduced how to select cross-chain nodes based on reputation. Furthermore, we provide a detailed introduction to

the process of using BLS signatures to verify data in Sect. 4. The experiment results are stated in Sect. 5. Finally, Sect. 6 concludes this article.

2 Related Work

In this section, we introduce the research on blockchain in IoT and cross-chain technology.

2.1 Blockchain in IoT

Existing work applies blockchain to IoT. Blockchain can enhance the security of IoT systems. With encryption and digital signatures by cryptographic keys [7], IoT data can be protected through blockchain. And the smart contract carried by the blockchain can automatically update the firmware of IoT devices and close the vulnerabilities that are susceptible to attacks [4]. Moreover, IoT data stored on the blockchain data can be identified and verified anywhere and anytime. For example, the work of Lu et al. [11] develops a blockchain-based product traceability system that provides traceability to suppliers and retailers. In this way, the quality and originality of products can be checked and verified. Boudguiga et al. [3] proposed a decentralized mechanism to push updates to IoT devices using blockchain. The blockchain is used to record transactions for software updates pushed to the device to prevent malware updates on the device. In this case, there is no need for a trusted agent to deliver the updates as the updates propagated to the device through the blockchain have guaranteed integrity.

The diversity and heterogeneity of IoT devices pose a huge challenge for blockchain [12]. Optimization of blockchain networks is one way to address performance. In 2021, Zhou et al. [17]. proposed an optimization mechanism in resource-constrained IoT systems. They improve the performance of the system by dynamically adjusting optimal block assignments. Zhang et al. [16]. analyze the IoT traffic by establishing a blockchain network that matches the scale of IoT. And they implement a lightweight Bitcoin-like blockchain based on PoW to solve the problem of high traffic load and network congestion.

2.2 Cross-Chain Technology

Cross-chain technology was first applied to the exchange of assets. It is mainly used for the conversion of Bitcoin and Ethereum. In 2014, adam et al. propose sidechain [2], which is a blockchain system independent of bitcoin. Sidechain can access the Bitcoin network and interact with the Bitcoin ledger to enable asset transfers. As a separate blockchain, the technical solutions and consensus mechanisms adopted by sidechain are not restricted by the main chain. The notary schemes [9] is currently the most widely used cross-chain scheme. It set a trusted node in the blockchain system, which is responsible for completing cross-chain operations. However, the notary scheme uses a fixed node for cross-chaining,

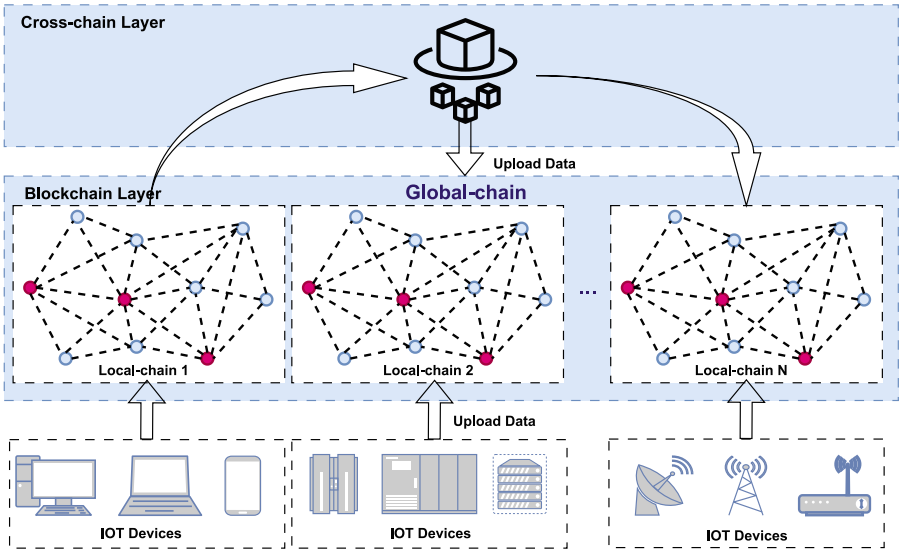


Fig. 1. System model.

which causes it to be more centralized. Once the node itself carries malicious intent or suffers an attack. It will not be able to guarantee the authenticity and trustworthiness of the cross-chain data. However, the cross-chain purpose of this scheme lies in asset transfer, without realizing a universal cross-chain approach.

Sun et al. proposed a decentralized cross-chain scheme [13], which combines a notary mechanism and hash-locking. By setting up multiple notaries and establishing an election mechanism, the degree of centralization of the cross-chain process is reduced. However, the cross-chain purpose of this scheme lies in asset transfer, without realizing a universal cross-chain approach. Ghosh et al. [8] proposed a decentralized gateway architecture that connects private blockchains to end users. The gateway employs a collective signature technique [14] to verify the data. However, this method allows only one-way communication and does not verify the identity of the requester.

3 System Architecture

In this section, we introduce the architecture and components of the system. As shown in Fig. 1, we build a multi-chain network structure in HyperLedger Fabric: including multiple local-chains, a global-chain, and a module that provides cross-chain functionality. IoT devices join the local-chains according to their different features and participate in the whole blockchain system. Next, we will introduce each component of the system in detail.

1. **IoT device:** IoT devices include all networked devices that have a need to participate in the blockchain. They upload important data to the blockchain,

such as identity information of personal computers, user access information in gateways, etc. Due to the complexity and large number of IoT devices and the heterogeneous nature of IoT data. Adding all IoT devices to the same blockchain will be a very difficult behavior. In this regard, we divide the devices according to their types and organize the IoT devices of the same type to join their respective local-chains.

2. **Local-chain:** The local-chain connects all IoT devices with similar features, such as personal computers and mobile phones that belong to the same smart device. The whole system will have multiple local-chains. It mainly accomplishes the information storage function of the system and is responsible for the data recording, identity granting and world state updating tasks within the local-chains. Taking the first local-chain in Fig. 1 as an example, it is responsible for storing user information, personal wallet, and other data uploaded by cell phones and computers onto the blockchain.
   Moreover, it also records the cross-chain requests and reputation scores of the devices, and these results will be recorded on the local-chain in a summarized form. IoT devices can selectively add one or more local-chains according to their geographic locations and device characteristics.
3. **Global-chain:** There is one and only one global-chain in the system. All blockchain nodes are to be added to the global-chain. The global-chain has two main functions. One is to store the local-chain data abstracts and provide validation function for the local-chain data. When a blockchain node submits data to the local-chain, it can choose to upload the abstracts to the global-chain. Unlike the local-chain which stores a large amount of data, the global-chain only needs to access a small amount of summary information. Second, it provides information records during cross-chain interaction. During cross-chain interaction, some parameters and key information of data exchange will be submitted to the global-chain to ensure security.
4. **Cross-chain module:** The cross-chain module is an important component in linking all local-chains. In this module, we propose a decentralized cross-chain verification scheme. The scheme consists of two parts. The first one is a reputation-based node selection mechanism. We calculate the reputation value of a node based on its network latency and the number of times it has been invoked. Multiple nodes with a good reputation are selected to participate in cross-chain verification. This solves the centralization problem under the notary schemes. It can effectively avoid a single point of failure and improve cross-chain security.
   The second is the data verification technology based on BLS signature. In the cross-chain process, the selected nodes will utilize the BLS signature to sign and verify the data. After that, the node with the highest reputation utilizes BLS to aggregate the signatures of each node to jointly complete the verification of the data. The specific cross-chain process will be introduced in the next section.

4 Cross-Chain Scheme

In this section, we introduce the cross-chain scheme proposed in this paper. It includes a cross-chain interaction process based on BLS signatures and a Reputation-Based Node Selection mechanism.

4.1 Cross-Chain Process Based on BLS Signature

The verification process based on the BLS signature consists of BLS signature algorithm and Shamir based secret sharing algorithm. BLS signature is used to avoid excessive storage cost of the final combined uplink signature, while Shamir secret sharing algorithm is used to achieve a certain number or more node endorsements for specific cross-chain transactions on the basis of BLS signature, thereby ensuring security.

As shown in Fig. 2, our cross-chain process consists of 6 stages: Request, Key Generation, Signature, Aggregation, Response, and Verification. Next, we will describe the 6 stages in detail.

Request: The request stages consists of 2 steps: main steps. First, node B_i of B initiates a request $Cross-chain.request$. Smart contract B_cross receives and parses the request. It will obtain the address of the other party of the cross-chain. After that, it forwards the request to A.

Key Generation: This stage also consists of 2 steps: Setup and Generation.

step1: $Setup(\lambda) \rightarrow \{G_1, G_2, G_T, e, g_1, g_2, p, h\}$

In the setup step, the smart contract B_{Cross} first parses the request. Determine the number of nodes participating in the cross-chain: n. And select the n nodes with the highest reputation. This includes A_j, A_x, A_y, and A_z where A_j is the node with the highest score. After that, B_Cross sends the parameters to the key generation center(KGC).

Then KGC obtains the relevant security parameters p of the algorithm and the elliptic curve bilinear mapping functions $e : G_1 \times G_2 \rightarrow G_T$. Multiplicative Cyclic group G_1, G_2 and its generator g_1, g_2, hash function h for mapping points onto elliptic curves. Specifically, Algorithm Setup(λ) inputs security parameters λ, and outputs as Two multiplicative Cyclic groups of a prime p: G_1, G_2 and a Bilinear map $e : G_1 \times G_2 \rightarrow G_T$. And g_1, g_2 as the generator of G_1, G_2. At the same time, a hash function is used in the scheme to map the hash digest of the signature data to the elliptic curve. These parameters are publicly available in the network:

$$h : \{0,1\}^* \rightarrow G_1 \tag{1}$$

step2: $Generate(G_2, p, t, n) \rightarrow \{MSK, MPK, SK, PK\}$

After initialization of the relevant parameters, KGC generates the master private key MSK, the master public key MPK, the threshold private key set

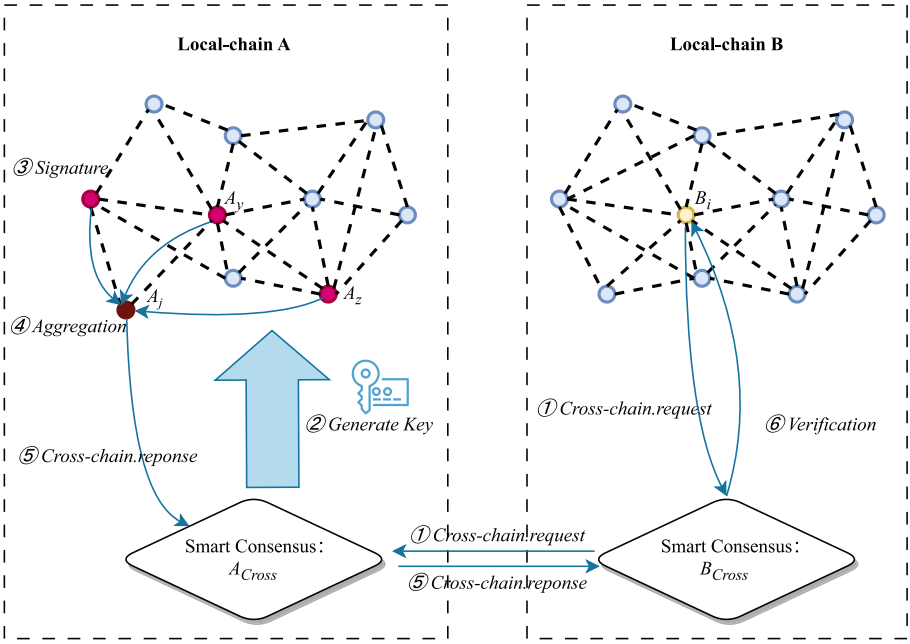


Fig. 2. Cross-chain process of DECS.

SK, and the threshold public key set PK. After the generation is complete, the generation center discloses the master public key MPK and the user group's public key PK. The key generation center randomly selects a random integer as shown in the Eq. (2)

$$x \leftarrow Z_p^* \tag{2}$$

At the same time, KGC set it as the master private key $MSK = x$, and obtains the master public key according to Eq. (3), where G is a randomly selected point from the elliptic curve:

$$v \leftarrow x \times G \in G_2 \tag{3}$$

After this, v will be set as the master public key $MPK = v$. Subsequently, the key generation center randomly selects $t-1$ elements $a_1, a_2, a_3, \ldots, a_{t-1} (a_i \in Z_p^*)$, and set $a_0 = x$, thus constructing a polynomial of order $t-1$:

$$f(x) = \sum_{i=0}^{t-1} a_i x^i \tag{4}$$

Then, KGC calculates x for each participant i $x_i = f(i)$ and set the corresponding threshold private key $sk_i = x_i$. Calculate v_i also according to Eq. (3). And set $pk_i = v_i$. It also calculates for n participants to obtain $SK = x_1, x_2, x_3, \ldots, x_n$ and $PK = \{v_1, v_2, v_3, \ldots, v_n\}$.

After the computation is completed, KGC sends the threshold encryption private key to the selected nodes through a secure channel.

Signature: $Sign(SK, m, t, h) \rightarrow \sigma_i$

Each participant i receives the private key and signs the data $m = \{0,1\}^*$. First, participant i hashes the data and maps the resulting hash digest to G_1: $h(M) \in G_1$, and $M = m||d$, with d equal to 0, 1, 2, ... This is because if the value obtained by hashing m is mapped directly onto the curve, there is a 50% probability that it will not map to a particular point. Therefore, the value of d is increased until the point is successfully mapped. The signature σ_i of participant i will be as shown in Eq. (5:)

$$\sigma_i \leftarrow x_i \times h(M) \in G_1 \tag{5}$$

Aggregation: $Aggregate(\sigma) \rightarrow \sigma$

The node A_j with the highest reputation score is responsible for aggregating signatures. It collects signatures from n participants. When it receives a signature group σ generated from the set Q combined by t participants. and the individual signatures within the signature group are verified. A_j can obtain the signature of the master private key MSK on the data based on Lagrange interpolation.

$$\begin{aligned}
\sigma &= \sum_{i \in Q} \sigma_i \prod_{j \in Q, j \neq i} \frac{j}{j-i} \\
&= \sum_{i \in Q} (h(M)) \times x_i \prod_{j \in Q, j \neq i} \frac{j}{j-i} \\
&= h(M) \times \sum_{i \in Q} x_i \prod_{j \in Q, j \neq i} \frac{j}{j-i} \\
&= h(M) \times x
\end{aligned} \tag{6}$$

Response: After signature aggregation, A_j packages the cross-chain data and MSK in Cross-chain.response. Then A_j sends the response data to B_i via A_{Cross}.

Verification: $Verify(MPK, \sigma, h(M), G, e) \rightarrow true|false$

After the B_i obtains the complete signature obtained by the endorsement initiator, the signature can be verified based on the properties of the bilinear function. The specific principle of verification is shown in the following equation:

$$\begin{aligned}
e(\sigma, G) &= e(x \times h(M), G) \\
&= e(h(M), x \times G) \\
&= e(h(M), MPK)
\end{aligned} \tag{7}$$

If the final calibration determines that $e(\sigma, G) = e(h(M), MPK)$ is equal, then the returned output is true, and the opposite is false.

The above six steps describe the cross-chain process of the BLS-based threshold signature scheme. The scheme utilizes a polynomial to hide the master private key in the BLS signature method and generates the private keys of the participants from it. Then a specific number of individual participant signatures on the data are integrated using an elliptic curve bilinear mapping function. The master signature is recovered using Lagrange interpolation. Validation is then performed to determine the validity of the transaction.

4.2 Reputation-Based Node Selection Mechanism in Cross-Chain

Parameter Settings: Before describing this mechanism, We first define the following.

Definition 1: Assuming that there are m nodes within a localized chain. One of the nodes is denoted as v_i. where $1 \leq i \leq m$. Each node maintains a called coefficient c_i. And the initial value is c_Init, which satisfies the following equation:

$$c_{init} = \frac{1}{m} \tag{8}$$

Definition 2: A node that is not part of the current local-chain is selected as a cross-chain requester. It records the network latency t for each node in $V = \{v_1, v_2, ..., v_m\}$. The time factor r_i is then computed for each v_i. r_i satisfies the following equation:

$$r_i = \frac{t_i}{\sum_{j \in V} t_j} \tag{9}$$

Definition 3: Knowing the called coefficient c_i and the time coefficient r_i of a node v_i, the cross-chain initiator determines the weight parameter $\alpha(\alpha \in [0, 1])$, for which it can compute the prestige value p_i, which satisfies the following equation:

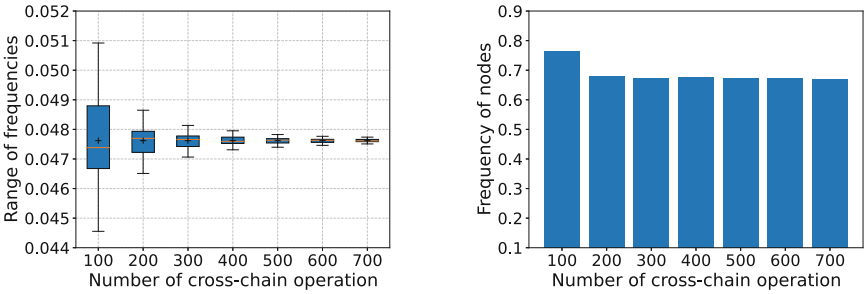
$$p_i = \alpha c_i + (1 - \alpha) r_i \tag{10}$$

Definition 4: After each cross-chain completion, each selected node v_i needs to be updated with the following equation:

$$c_i = c_i + \frac{1}{mn} (i \in N) \tag{11}$$

And the unselected node v_i also needs to update with the following equation:

$$c_i = c_i - \frac{1}{m(m-n)} (i \in V and i \notin N) \tag{12}$$



(a) Range of all node frequencies vs. Number of cross-chain

(b) Frequency of the 8 nodes with the lowest latency vs. Number of cross-chain

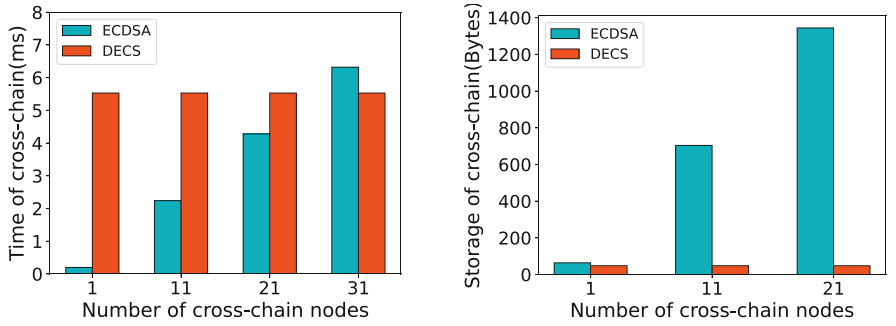
Fig. 3. Frequency of nodes participating in cross-chain.

Reputation Update Process: We use smart contracts to automatically update the reputation scores of nodes. The core idea of this contract is to consider the frequency of participation in cross-chaining and network latency of each node, achieving a balance between frequency and network latency. After the node participant in the cross-chain, its called coefficient c_i is updated. The smaller the c_i is, the more likely it is to be selected as a participant for cross-chain. The specific description is as follows:

1. After the blockchain network is initialized, the smart contract awards cross-chain participation status to the nodes with high trustworthiness. And initialize their called coefficients $c_i = c_(init)$.
2. Cross chain initiator s_r is for each strong node v_i Calculate reputation p_i. And sort the obtained values in ascending order, determine the priority of the signature, select nodes in order to form a set N, send (t, n) threshold signature requests, and send unselected messages to nodes outside the set N.
3. After collecting c_i, t_i from these nodes, the smart contract computes the reputation p_i for each node v_i. And it sorts the obtained values in ascending order. And select the nodes in order to combine them into a set N. A (t, n) threshold signature request is sent to the nodes in the set, and an unselected message is sent to nodes outside the set N.
4. Node $v_k(k \in N)$ that receives threshold signature request performs the reputation update algorithm of the Eq. (11). And the unselected nodes $v_l(l \in Vandl \notin N)$ perform the reputation update algorithm of the Eq. (12).

5 Performance Analysis

We tested our scheme on the HyperLedger Fabric, using Intel (R) Xeon (R) Silver 4214 CPU with 256 GB RAM with 16TB hard drive. We built a blockchain network with 21 nodes. It includes two local-chains and one global-chain. One of



(a) Time cost of cross-chain vs. number of closs-chain nodes

(b) Storage cost of cross-chain vs. number of closs-chain nodes

Fig. 4. Time and Storage cost of cross-chain.

the local-chains has 9 nodes and the other has 12 nodes. The two local-chains are independent of each other. The blockchain network adopts the Raft consensus, with a maximum blocking time of 7 s, a maximum number of 100 transactions, and a maximum block size of 100M. At the same time, the experiment used the mainstream testing tool Caliper of the Hyperledger Fabric to test blockchain performance.

Frequency of Nodes Participating in Cross-chain: The first experiment targets the Reputation-Based Node Selection Mechanism to test whether the selection of cross-chain nodes is decentralized. We initiate a cross-chain request to a local-chain of 12 nodes. And set the number of cross-chain nodes to 8, and the reputation parameter $\alpha = 0.5$, to test the frequency of each node participating in the cross-chain.

The experimental results are shown in Figs. 3a and Figs. 3b. Figures 3a is a box plot of the frequency of all nodes participating in cross-chaining. It can be seen that there is still an obvious gap in the frequency of the nodes when completing 100 cross-chaining. And as the number of cross-chaining increases. The gap in frequency gradually decreases. Each node can be selected evenly. Figures 3b count the 8 nodes with the lowest network latency. And calculate the sum of their frequencies. It can be seen that at 100 experiments, the frequency is the highest at 0.76. With the increase in the number of experiments, the value gradually decreases and reaches a stable value of 0.67. This number is the ratio of the number of nodes to the number of all nodes. It shows that our scheme is able to decentralize the selection of nodes participating in cross-chain.

Time and Storage Cost of Cross-chain: We compare the DECS proposed in this paper with the ECDSA-based notary scheme [9]. Compare the time and storage cost used to validate cross-chain data for both. First, We performed a step-by-step test of BLS signatures. The experimental results are shown in the Table 1. Each step was experimented with 100 times and averaged.

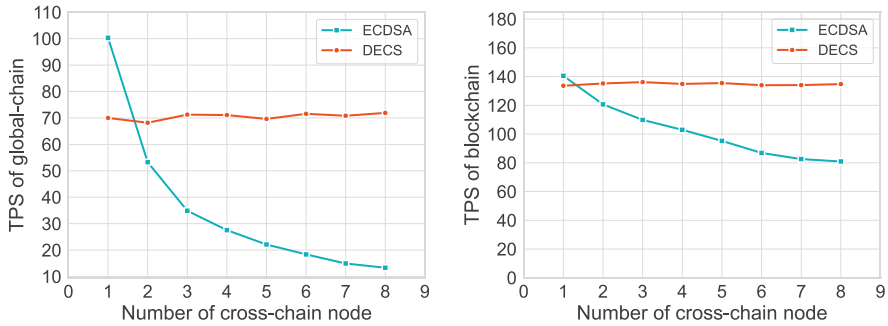
Table 1. Time cost for each step of the BLS signature.

Notation	Description	execution time
T_z	Take a random number in T_z	0.0183 ms
T_{G_1}	Point multiplication in G_1	0.5482 ms
T_{G_2}	Point multiplication in G_2	0.6671 ms
T_a	Point addition in G_1	0.3030 ms
T_p	Polynomial calculation time	0.0053 ms
T_h	Hashing with SHA256	0.0027 ms
T_e	Verification of Bilinear Functions in T_z	3.9822 ms

Figure 4a shows the time cost required by the two schemes. The experiment starts with the most basic notary scheme, which means that only 1 node is involved in the cross-chain. We can see that at this point the ECDSA scheme significantly outperforms the DECS scheme. However, as the number of cross-chain nodes increases, the time spent by the ECDSA scheme keeps increasing. This is because the ECDSA scheme is unable to aggregate signatures. The verifier needs to verify all the signatures one after another. On the other hand, the DECS scheme can keep the verification time at 5.53 milliseconds due to the aggregation of signatures.

Figure 4b illustrates the storage cost required by both schemes. Similar to the time cost. The ECDSA scheme has increasing storage space as the number of cross-chain nodes. In contrast, the DECS scheme only has one MPK for each cross-chain process, regardless of the number of cross-chain nodes. So the storage cost remains constant. Since this scheme uploads the public key into the global-chain, the storage size will significantly affect the overall performance of the blockchain. This gives the DECS scheme a more obvious advantage.

Blockchain Performance: Figure 5a and Fig. 5b illustrates the TPS of the blockchain under both scheme. Where Fig. 5a shows the TPS of the global-chain. As the number of cross-chain nodes increases. The data size that needs to be stored on the global-chain increases for the ECDSA scheme. The throughput of the blockchain also keeps decreasing. The same effect is seen in Fig. 5b. Figure 5b shows the test performed for the entire blockchain network, which includes two local-chains and one global-chain. As we can be seen from the figure, the performance of the ECDSA scheme keeps decreasing. While the performance of the DECS scheme remains stable in both experiments. When the number of cross-chain nodes reaches 8, the DECS scheme significantly outperforms the ECDSA scheme.



(a) TPS of global-chain vs. Number of cross-chain nodes.

(b) TPS of entire blockchain vs. Number of cross-chain nodes.

Fig. 5. Blockchain performance comparison.

6 Conclusion

In this paper, we present a novel decentralized and efficient cross-chain scheme tailored for IoT systems. Our constructed system demonstrates excellent adaptability to the diverse and intricate nature of IoT environments. Furthermore, our proposed cross-chain solution effectively addresses the pervasive issue of excessive centralization within current cross-chain technologies. The incorporation of BLS signature-based data verification alleviates the burden of high storage requirements associated with the ECDSA scheme. Experimental results show that our scheme has better decentralization and efficiency, and the blockchain throughput has a good performance. We hope our scheme can be used as a high-performance tool for IoT systems to provide efficient, secure protection for IoT data.

References

1. Ahluwalia, S., Mahto, V, R., Guerrero, M.: Blockchain technology and startup financing: a transaction cost economics perspective. Technol. Forecast. Soc. Change **151** (2020). https://doi.org/10.1016/j.techfore.2019.119854
2. Back, A., et al.: Enabling blockchain innovations with pegged sidechains. **72**, 201–224 (2014)
3. Boudguiga, A., et al.: Towards better availability and accountability for IoT updates by means of a blockchain. In: 2017 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), pp. 50–58 (2017). https://doi.org/10.1109/EuroSPW.2017.50
4. Christidis, K., Devetsikiotis, M.: Blockchains and smart contracts for the internet of things. IEEE Access **4**, 2292–2303 (2016). https://doi.org/10.1109/ACCESS.2016.2566339

5. Dai, H.N., Zheng, Z., Zhang, Y.: Blockchain for internet of things: a survey. IEEE Internet Things J. **6**(5, SI), 8076–8094 (2019). https://doi.org/10.1109/JIOT.2019.2920987
6. Deng, L., Chen, H., Zeng, J., Zhang, L.-J.: Research on cross-chain technology based on sidechain and hash-locking. In: Liu, S., Tekinerdogan, B., Aoyama, M., Zhang, L.-J. (eds.) EDGE 2018. LNCS, vol. 10973, pp. 144–151. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-94340-4_12
7. Genc, Y., Afacan, E.: Design and implementation of an efficient elliptic curve digital signature algorithm (ecdsa). In: Chakrabarti, S., Paul, R., Gill, B., Gangopadhyay, M., Poddar, S. (eds.) 2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), pp. 1026–1031 (2021). https://doi.org/10.1109/IEMTRONICS52119.2021.9422589. IEEE; Inst Engn & Management; IEEE Vancouver Sect; IEEE Toronto Sect; SMART; Univ Engn & Management , iEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), ELECTR NETWORK, APR 21-24, 2021
8. Ghosh, B.C., Bhartia, T., Addya, S.K., Chakraborty, S.: Leveraging public-private blockchain interoperability for closed consortium interfacing. In: IEEE INFOCOM 2021 - IEEE Conference on Computer Communications, pp. 1–10 (2021). https://doi.org/10.1109/INFOCOM42981.2021.9488683
9. Hope-Bailie, A., Thomas, S.: Interledger: Creating a standard for payments. In: Proceedings of the 25th International Conference Companion on World Wide Web, pp. 281–282. WWW '16 Companion, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE (2016). https://doi.org/10.1145/2872518.2889307
10. Kang, E.S., Pee, S.J., Song, J.G., Jang, J.W.: A blockchain-based energy trading platform for smart homes in a microgrid. In: Proceedings of 2018 3rd International Conference on Computer and Communication Systems (ICCCS), pp. 472–476 (2018). IEEE; Nagoya Inst Technol, 3rd International Conference on Computer and Communication Systems (ICCCS), Nagoya, JAPAN, APR 27-30, 2018
11. Lu, Q., Xu, X.: Adaptable blockchain-based systems a case study for product traceability. IEEE Softw. **34**(6), 21–27 (2017). https://doi.org/10.1109/MS.2017.4121227
12. Makhdoom, I., Abolhasan, M., Abbas, H., Ni, W.: Blockchain's adoption in IoT: the challenges, and a way forward. J. Netw. Comput. Appl. **125**, 251–279 (2019). https://doi.org/10.1016/j.jnca.2018.10.019
13. Sun, Y., Yi, L., Duan, L., Wang, W.: A decentralized cross-chain service protocol based on notary schemes and hash-locking. In: 2022 IEEE International Conference on Services Computing (SCC), pp. 152–157 (2022). https://doi.org/10.1109/SCC55611.2022.00033
14. Syta, E., et al.: Keeping authorities honest or bust with decentralized witness cosigning. In: 2016 IEEE Symposium on Security and Privacy (SP), pp. 526–545 (2016). https://doi.org/10.1109/SP.2016.38
15. Tschorsch, F., Scheuermann, B.: Bitcoin and beyond: a technical survey on decentralized digital currencies. IEEE Commun. Surv. Tutor. **18**(3), 2084–2123 (2016). https://doi.org/10.1109/COMST.2016.2535718

16. Zhang, H., Lao, L., Shu, C., Xiao, B.: Analysis of the communication traffic model for permissioned blockchain based on proof-of-work. In: IEEE International Conference on Communications (ICC 2021). IEEE International Conference on Communications, IEEE (2021). https://doi.org/10.1109/ICC42927.2021.9500333, Telus; Huawei; Ciena; Nokia; Samsung; Qualcomm; Cisco; Google Cloud, iEEE International Conference on Communications (ICC), ELECTR NETWORK, JUN 14-23, 2021
17. Zhou, J., Feng, G., Wang, Y.: Optimal deployment mechanism of blockchain in resource-constrained IoT systems. IEEE Internet Things J. **9**(11), 8168–8177 (2022). https://doi.org/10.1109/JIOT.2021.3106355