The Journal of Supercomputing (2022) 78:16266–16285
https://doi.org/10.1007/s11227-022-04528-x

GBRM: a graph embedding and blockchain-based resource management framework for 5G MEC

Kai Lei[1] · Hao Ye[1] · Junjie Fang[1] · Peiwu Chen[2] · Liangjie Zhang[3] · Jing Xiao[2]

Accepted: 30 March 2022 / Published online: 3 May 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract
In the 5G scenario of the convergence of information technology (IT) and communication technology (CT), multi-operators collaborate to form edge computing, which makes the problem of resource optimization more complicated than ever. Users may access resources deployed by various MEC's operators to achieve ultra-low latency. However, traditional resource management methods consider only a single operator failure to handle profit allocation and privacy security issues among different operators. To address this problem, we proposed a resource management framework named GBRM based on graph embedding and blockchain. Specifically, we use the Stackelberg game model to solve MEC servers' cache-offloading problem; non-indexed content sharing by Deepwalk graph embedding between MECs ensures the privacy of different operators' content. Consortium blockchain assists in the trusted profit allocation of services across various operators. Experiments show in the virtual network scenario that our work performance is significantly better than the RandomSelect and the LocalIndex method in global latency and close to the global index's ideal situation. Multi-operators collaborate to form edge computing, which makes the problem of resource optimization more complicated than ever.

Keywords 5G · Mobile edge computing · Graph embedding · Blockchain · Stackelberg game · Resource management

1 Introduction

In the 5G era, information technology (IT) and communication technology (CT) are integrated together. Mobile edge computing (MEC) is one of the main approaches that enable the integration of IT into the telecommunication infrastructure [1]. MEC implements the cloud computing paradigm to the network's edge, especially the

✉ Kai Lei
leik@pkusz.edu.cn

Extended author information available on the last page of the article

radio access network (RAN). By equipping the edge network infrastructures with enhanced computing and caching capabilities, many applications such as the Internet of Things (IoT) and large scale video content distribution with stringent latency, scalability and throughput requirements will benefit from it.

We consider a typical 5G MEC scenario MEC servers belonging to multiple Internet Service Providers (ISP) provide network access, computing tasks, cached content and other services for end-users and content providers, where users may access resources deployed by other operators. Generally, MEC servers refer to the Internet infrastructures such as 5G base stations deployed by certain Internet Service Providers (ISP). However, MEC's vision is to utilize all possible spare computing and caching resources of edge devices owned by ordinary users or particular infrastructure providers. Furthermore, 5G requires base stations to be deployed in an ultra-dense fashion. Therefore, we consider a future vision and refer MEC servers to macro-base stations deployed by different stakeholders. Apart from providing network access services to end-users, MEC servers cache content for content providers (CPs) to reduce average download delay and backbone pressure.

Three challenges remain to be tackled to construct a multi-stakeholders 5G MEC system for content caching and distribution. Firstly, due to the limitation of caching resources of MEC servers and the dynamics of content request distribution, it is challenging to design a caching scheme for multiple CPs and multiple MEC servers. Secondly, considering that one 5G end-user may attach to multiple MEC servers and one content may exist in multiple MEC servers and CPs, there is a content source recommendation problem for end-user requests. Thirdly, MEC servers are deployed by different stakeholders and end-users, which belong to one specific stakeholder. They may utilize the MEC server deployed by other stakeholders as a network access point. Therefore, there exists a cross stakeholder settlement problem. We propose a GBRM framework for solving the above problem.

We consider a continuous caching decision-making process among multiple CPs and multiple MEC servers. CPs can lease caching resources from MEC servers with a specific caching incentive price. After being informed of the offered caching incentive price, MEC servers decide on the caching amount allocated to this CP. The caching incentive price decision of CPs is made based on the estimation of MEC servers' reaction, while the caching amount decision of MEC servers is made based on the observation of CPs' action. This continuous caching decision-making process is entirely under the Stackelberg game. Therefore, we propose a Stackelberg game-based caching scheme with CPs acting as leaders and MEC servers acting as followers. Furthermore, the benefits of CPs and MEC servers are defined separately as utilities. Stackelberg equilibrium at which point both CPs and MEC server get their maximum utilities is proved by mathematical analysis and simulation results.

After the Stackelberg game solves the problem of caching decision-making, the remaining issue is how to select the best MEC when the user requests content when the base station(BS) connects to multiple MECs. A naive but direct way is that the MEC connected to the base station sends the index of the content it contains to the MEC connected to it, and then, the BS uses these indexes to find the best MEC server. However, the BS will connect to the other operator's MEC in the 5G scenario who may not be willing to share their MEC content index due to privacy and security considerations; at the same time, the MEC may have a lot of content, causing the content index too large, sharing all content indexes may cause traffic load pressure on the network. To solve the above problem, we introduced a method based on embedding matching to reduce the amount of index transmission while preserving MEC content's privacy. The MEC content is formed into embedding and then combined to form the embedding of MEC send to the BS, and the BS measures the similarity between the content request and the embedding of the MEC to calculate the MEC is most likely to contain the request's content for access. Experiments show that GBRM performance is much better than random selection and is close to the traditional index method.

There remains a problem when MEC servers provide service (e.g., provide their cached content resources) to the users. Different operator's own MEC servers may serve customers who are not their own operators. Therefore, the base station may not be willing to do these things if they cannot benefit from these services. A reasonable solution to this problem is charge by service. Specifically, MEC servers will get paid each time they provide service whether the customers belong to their operator or not. There will be a system between the operators handling the cross-operator service cost. After all, the users do not care who provides the service; they still only pay their own operators, which share a portion of their revenue with MEC servers that serve their customers. Therefore, we have introduced blockchain to build the system for service cost settlement across operators. The scenario we are dealing with is a large number of untrusted or partly trusted nodes. MEC servers may lie to earn more benefits, and there is no supernode between operators to verify the authenticity or guarantee safe dealings of the MEC server's service. In a word, the aiming is to realize a reliable and efficient settlement beyond the distributed peer by its characters. The main contributions of this paper lie in:

- We propose a multi-stakeholders 5G MEC resource management framework named GBRM for content caching and distribution with graph embedding and blockchain assisted. The interrelations among different stakeholders are elaborated.
- The interaction of CPs and MEC servers is modeled as a Stackelberg game-based caching decision-making process, the Stackelberg equilibrium of which is proved by mathematical analysis and simulation results.
- In consideration of the multiple access points and multiple content sources feature of 5G, we solve the end-user requests and content sources matching problem using the graph embedding-based recommendation approach. This method is of high accuracy while protecting the operator's privacy.
- Since the MEC server will be deployed by different stakeholders in the future, there exists a problem of resolution between the untrusted stakeholders. We propose a consortium blockchain-based settlement scheme to encourage and enhance MEC servers' participation in this system.

The remainder of this paper is structured as follows. Section 2 gives a brief summary of related work in terms of Stackelberg game-based edge caching, graph embedding-based recommendation approach and blockchain and consensus. Section 3 presents the multi-stakeholders 5G MEC system model. Then, the following three sections elaborate on the design details of our Stackelberg game-based caching scheme, graph embedding-based recommendation approach and blockchain-based settlement scheme, respectively. Section 7 presents the simulation results and analysis. We conclude this paper in Sect. 8.

2 Background and related work

2.1 Stackelberg game-based edge caching

Many research effort has been dedicated to the edge caching problem, among which Stackelberg game-based ones are listed below. Li et al. [2] considers an edge caching scenario with several video retailers (VRs), network service providers (NSP) and several mobile users. The NSP acts as a leader and sets the price of leasing small-cell base stations (SBS), while the VRs act as followers and compete to rent a fraction of the SBSs. In [3], an edge caching scenario with one content provider (CP), one Internet Service Provider (ISP) and several users is considered. Furthermore, the Stackelberg Game-based solutions are listed below caching scheme is reversed with CP acting as a leader to offer to cache incentive price and ISP acting as a follower to decide on the caching amount allocation. In [4], the Stackelberg game-based caching scheme is decomposed into two kinds of sub-games, a storage allocation game (SAG) and multiple user allocation games (UAGs). Special attention is paid to the scalability issue. Xiong et al. [5] investigated a sponsored and edge caching content service market model. A three-stage Stackelberg game is formulated to jointly optimize each content service provider and mobile users' benefits.

2.2 Graph embedding-based recommendation

In the early recommendation system, collaborative filtering [6] is usually used to recommend content. First, the connection between the user and the content is formed into an adjacency matrix, and then, it is decomposed into two low-rank matrices of the user and the content through the method of matrix decomposition. However, this kind of approach faces a cold start and high computational complexity (the complexity of matrix factorization is $O(n^3)$). With the development of deep learning, DeepWalk [7] was proposed as a new method of graph embedding. Deep walk borrowed the training method in word2vec [8] and first used the random walk to traverse the graph for generating a walk sequence similar to a language sentence; a skip-gram is adopted to minimize the distance of similar nodes to generate embedding for the nodes. Some subsequent work improved the similarity calculation, such as Line [9] change the first-order similarity of Deepwalk to second-order or even higher-order similarity; some work improved the random walk method to consider various types of node features and extract more comprehensive information of the graph, such as metapath2vec [10] and HeteSpaceyWalk [11] for solving heterogeneous network; the paper BiNE [12] and BiANE [13] for solving the problem in Bipart network.

2.3 Blockchain and consensus

Blockchain is known as Bitcoin's underlying technology, which can be regarded as a shared digital ledger. The records of the transactions are stored in blocks, which are linked in chronological order. Each node in the blockchain network all has a copy of the ledger and synchronizes it regularly [14].

Blockchain is widely used to keep records, especially transaction records, in a distributed environment. It is due to non-tampering nature. Notably, each block is encrypted using the hash algorithm. Any change in input information will result in a significant change in the hash result and easily verified by all nodes.

If new blocks are trying to add to the blockchain's end, they need to pass the consensus across the entire networks' nodes. In our view, a consensus mechanism must address two issues: Firstly, Which node is selected to add a new block to the blockchain during this period? This can also be understood as a matter of avoiding forks by ensuring that all nodes keep the same copy of the blockchain in a distributed environment; secondly, how to ensure that the blockchain records are real and effective? For example, in Bitcoin, each node must query the blockchain to verify that one transaction generator has enough money to complete the transaction.

The proof of work (PoW) consensus [14] mechanism is the broadest deployed consensus mechanism in existing blockchains, which was introduced by Bitcoin. Its idea can be summarized as follows: All nodes try to solve a mathematical problem with a certain difficulty nonce, and the node which gets the answer first will get the right to add a block to the blockchain and the Bitcoin reward. One thing that can be inferred is that the more computational power one node has, the more likely it is to get the right to add a block.

Proof of Stake(PoS) [15, 16] is another widespread consensus applied in the blockchain. It has been seen as a strong candidate to replace the largely inefficient Proof of Work mechanism currently plugged in most existing open blockchains. In PoS, a node will not get the right to add a new block by solving the mathematical problem. The right is related to the proportion of coins held by a miner.

Proof of storage (PoS) is substantially not a consensus but an interactive cryptographic protocol that allows a client to verify that a server faith-fully stores a file efficiently [17, 18]. A blockchain focuses more on the second issue that a consensus must address–the records' authenticity and validity. The Filecoin has improved POS and implemented a distributed environment that pays for every node that helps users store files.

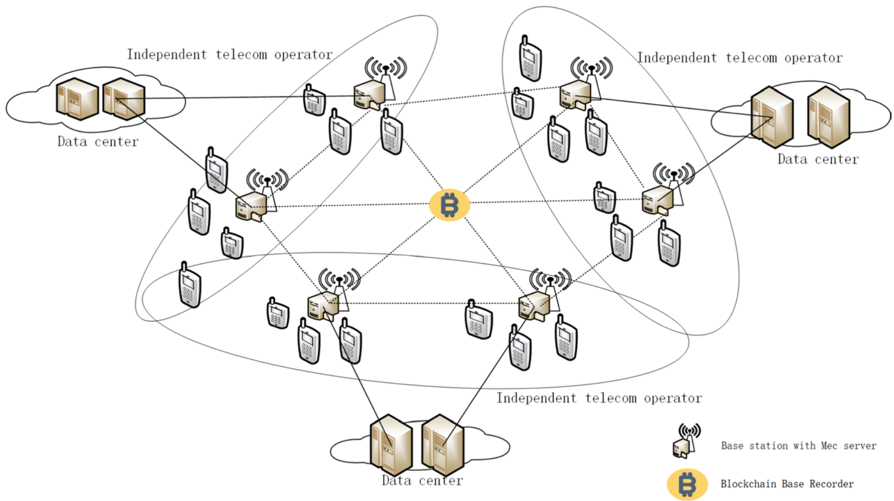


Fig. 1 5G MEC system model

3 System model

We consider a typical 5G edge computing scenario with M CPs, N FN(Fog node)s and G users, as shown in Fig. 1. The sets of CPs, FNs and users are denoted by $\{P_1, P_2, \dots, P_M\}$, $\{F_1, F2, \dots, F_L\}$ and $\{U_1, U_2, \dots, U_G\}$, respectively. FNs refer to 5G base stations, which are equipped with caching resources along with enhanced transmission capability that is densely deployed. Each user may be served by multiple FNs as long as it locates in an overlapping service area, which is a distinct feature of 5G. In the prospect of edge computing, FN may be deployed by any stakeholders, including Internet Service Providers, infrastructure manufactures and even common users.

CPs lease cache from FNs to cache popular content at the edge of the network to reduce average service delay. Due to the numerous amount of content, we divide content into K classes according to content popularity. The content class set is denoted as $\{f_1, f_2, \dots, f_K\}$. Suppose the amount of requests delivered by FN F_n to CP P_m for content f_k is λ_{nmk}. The caching incentive offered by CP P_m to FN F_n for caching a unit amount of content f_k is denoted as v_{mnk}. Let x_{nmk} denotes the amount of caching resource that FN F_n allocates to CP P_m for caching content f_k.

4 Caching scheme

In this section, a Stackelberg game-based caching scheme is proposed. We simplify the caching incentive and caching amount decision problem between multiple CPs and multiple FNs and consider the interaction between one CP and multiple FNs. Firstly, we define the utilities of each CP and each FN separately. The optimization goal of the caching scheme is to maximize the utilities of CP and FNs simultaneously. Then, considering the continuous decision-making process between CP and multiple FNs, we model the interaction between CP and multiple FNs as a Stackelberg game between one leader and multiple followers. Afterward, the Stackelberg equilibrium between CP and FNs is solved by mathematical analysis.

4.1 Utility definition

As for CP P_m, its utility equals income minus cost as shown in Eq. (1). Content fee from users is the major income source of CP. Suppose that the unit content fee is set as v_{con} and users request λ_{mk} chunks of content f_k from CP P_m, thus the income of CP equals to $\sum_{k=1}^{K} v_{con}\lambda_{mk}$. The cost of CP is composed of three parts including caching fee, transmission fee and users' dissatisfaction level. The caching fee that CP P_m pays to all FNs adds up to $\sum_{k=1}^{K} v_{mnk}x_{nmk}$. With v_{tra} denoted as the unit transmission fee, the transmission cost of CP P_m equals to $\sum_{k=1}^{K}\{v_{tra}(\lambda_{mk} - \sum_{n=1}^{N} x_{nmk})\}$. Users' dissatisfaction level depends on the average service delay and reflects the long-term profitability of CP. As shown in Eq. (1), users' dissatisfaction level is modeled as a non-decreasing function of the total caching amount, whose first derivative is non-increasing. S_{mk} is normalized into the interval [0, 1] and stays unchanged as 1 in the case of total caching amount exceeds request amount. g denotes the weight factor of users' dissatisfaction level in the utility of CP.

$$W_m(V_m) = \sum_{k=1}^{K}\{v_{con}\lambda_{mk} - Cost_{mk}\} \tag{1}$$

$$Cost_{mk} = \sum_{n=1}^{N} v_{mnk}x_{nmk} - v_{tra}(\lambda_{mk} - \sum_{n=1}^{N} x_{nmk}) - g(1 - S_{mk}) \tag{2}$$

$$S_{mk} = \begin{cases} \dfrac{-\frac{1}{S_F}(\sum_{n=1}^{N} x_{nmk})^2 + 2w_k\sum_{n=1}^{N} x_{nmk}}{w_k\lambda_{mk}} & \sum\limits_{n=1}^{N} x_{nmk} < \lambda_{mk} \\ 1 & \sum\limits_{n=1}^{N} x_{nmk} \geq \lambda_{mk} \end{cases} \tag{3}$$

As shown in Eq. (4), ISP P_m gets profit by providing caching services to CP and transmission service to CP and users. Let v_{tra} and C denote the unit transmission fee and cost, respectively. It is worth noting that the content transmitted for CP P_m

equals to cache miss amount $x_{nmk} - \lambda_{nmk}$. Therefore, the transmission profit equals to $\sum_{k=1}^{K}\{v_{tra} - C)(2\lambda_{nmk} - x_{nmk})\}$. $\sum_{k=1}^{K} v_{mnk}x_{nmk}$ denotes the caching reward of FN F_n which is the same as the caching cost of CP P_m. δ_{nk} denotes the caching cost of FN F_n for caching x_{nmk} chunks of content f_k, the definition of which follows the general rule of marginal cost.

$$W_n(X_n) = \sum_{k=1}^{K}\{(v_{tra} - C)(2\lambda_{nmk} - x_{nmk}) + v_{mnk}x_{nmk} - \delta_{nk}\} \tag{4}$$

$$\delta_{nk} = C_0 x_{nmk}(\frac{x_{nmk}}{\lambda_{nmk}})^{\theta} \tag{5}$$

4.2 Stackelberg game-based caching scheme

As caching scheme between multiple CPs and multiple FNs can be easily expanded to caching scheme between one CP and multiple FNs. We solve the caching amount and caching price decision problem between one CP and multiple FNs. In the interaction between CP and FNs, CP decides on the caching prices V_m to maximize its utility. Afterward, each FN makes decision on the caching amount X_n to maximize its utility. The caching price decision of CP is based on the estimation of FN's reaction. The caching amount decision of FN is based on the observation of CP's action. Based on the above analysis, we model the interaction between CP and FNs as a Stackelberg game with CP acting as a leader and FNs acting as followers. We aim at designing a caching scheme that solve the caching price and caching amount that maximize the utilities of CP and FNs simultaneously, which is the Stackelberg equilibrium between CP and FNs.

Since the action of CP is made based on the estimation of each FN's reaction. Therefore, we try to figure out the optimal reaction function of each FN. More specifically, given the caching prices offered by CP, what's the optimal caching amount to maximize the utility of FN. The second-order derivative of the utility of FN F_n with respect to x_{nmk} is shown in Eq. (6). On the condition of $\theta > 0$, the Hessian matrix is negative definite matrix. Therefore, the maximum utility of CP is obtained at setting the first-order derivative of W_n with respect to x_{nmk} as 0. The optimal reaction function of caching amount x^*_{nmk} with respect to caching price v_{mnk} is shown in Eq. (7).

$$\frac{\partial^2 W_n}{\partial x_{nmk}\partial x_{nmh}} = \begin{cases} -\frac{C_0\theta(\theta+1)}{(\lambda_{nmk})^{\theta}}(x_{nmk})^{\theta-1}, & k = h \\ 0, & k \neq h \end{cases} \tag{6}$$

$$x^*_{nmk} = \lambda_{nmk}[\frac{v_{mnk} + C - v_{tra}}{C_0(\theta + 1)}]^{\frac{1}{\theta}} \tag{7}$$

Replacing x_{nmk} in the utility of CP $W_m(V_m)$ with x^*_{nmk}, $W_m(V_m)$ remains the function of $V_m = \{V_{m1}, V_{m2}, ..., V_{mN}\}$. Since multiple variables remain to be resolved in one function, coordinate ascent algorithm is adopted to find the optimal V_m that maximizes the utility of CP. According to coordinate ascent algorithm, the initial value of $V_m = \{V_{m1}, V_{m2}, ..., V_{mN}\}$ is picked randomly. Then, update the value of each v_{nmk} according to the optimal update function supposing that other v_{nmk} are known. The iterative update terminates on condition that the gap of the value of $W_m(V_m)$ between two iterations is smaller than certain threshold. Therefore, we intend to find the optimal update function of each v_{nmk}. On the condition of the first-order derivative of $W_m(V_m)$ with respect to v_{mik} equal to 0, the second-order derivative of $W_m(V_m)$ with respect to v_{mik} is shown in Eq. (8). As long as $\lambda_{imk} > \frac{\theta}{3}$ and $g > 0$, the second-order derivative of W_m with respect to v_{mik} is negative, which means the maximum W_m is obtained when the first-order derivative of W_m with respect to v_{mik} equals to 0. The first-order derivative of W_m with respect to v_{mik} is shown in Eq. (9).

$$\frac{\partial^2 W_m}{\partial^2 v_{mik}} = \left(1 - \frac{3\lambda_{imk}}{\theta}\right)\left(\frac{x^*_{imk}}{\lambda_{imk}}\right)^{1-\theta} - \frac{2 * g * flag}{\theta^2}\left(\frac{x^*_{imk}}{\lambda_{imk}}\right)^{2-2\theta} \tag{8}$$

$$\frac{\partial W_m}{\partial v_{mik}} = -x_{imk} + \frac{\partial x_{imk}}{\partial v_{mik}}\left\{ v_{tra} - v_{mik} + 2g * flag * \left(\frac{1}{\lambda_{mk}} - \frac{\sum_{n=1}^{N} x_{nmk}}{\lambda^2_{mk}}\right)\right\} \tag{9}$$

$$flag = \begin{cases} 1, & \sum_{n=1}^{N} x_{nmk} < \lambda_{mk} \\ 0, & \sum_{n=1}^{N} x_{nmk} \geq \lambda_{mk} \end{cases} \tag{10}$$

5 Graph embedding-based MEC framework

The purpose of graph embedding is to design a framework to solve the end-users requests and content sources matching problem. As we described before, a 5G edge computing scenario may contain multiple CPs, FNs and G users. This chapter elaborates on this problem as a communication base station requests content for multiple MECs. Since in 5G scenarios, the user, mobile phone and IoT devices may be connected to multiple base stations. So we can solve the content source match problem in the communication base station. After the simplify, the problem becomes this: The communication base station receives the end-user device's request. Then, communication base station depends on a specific strategy to choose a MEC or central network to get the user request content. At last, the base station gets the content and returns what end-user needed. We list the matching strategy challenge below:

- When should a base station request content from MEC and when should it request from the central network? The base station may not know the MEC server's content, as the MEC content may holder by multiple stakeholders who are not want to share the content details they owned caused by the problem of privacy. Simultaneously, the item may also change quickly from time and the vast number of MEC leads the base station to save the content index impossible and impractical.

- How to ensure the service delay is as low as possible. If the base station chooses to connect to MEC to get the required content but failed and subsequently chooses to connect to the central network. The latency may too high to satisfy the end-user, or the base station flooding to choose the optimal MEC, the price may not be affordable. So we need a feasible method to choose the highest probability MEC, which meets the content demand.

To solve the above problem, we propose a graph embedding-based content recommended system. Firstly, we use a graph representation algorithm to represent the different contents in the MEC as embedding vector, and then, the MEC combines the different content embeddings as a single embedding vector represents the whole MEC content storage, at last, the communication base station calculates the similarity between the multiple MEC embedding and request content embedding for choosing the MEC includes the content with the highest probability. We split the framework into three parts: (1) content embedding, (2) MEC embedding, (3) graph embedding indexing, as shown in Fig. 2

5.1 Content embedding

The process to generate content embedding is to generate a global content index. As a content provider may contain various modal content, such as videos, images, and text, the request matching system should handle it with the same input to produce the same outputs in different MEC for consistency. We select a message-digest algorithm like MD5 [19] and SHA-256, which function generate equal length string

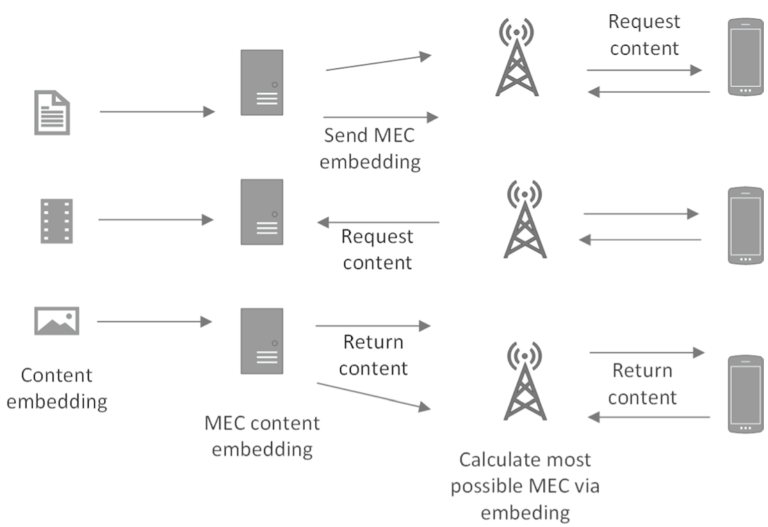


Fig. 2 Graph embedding base framework

output for arbitrary length input and reach our previous need with the same inputs produce the same outputs.

5.2 MEC embedding concat

The total amount of hash string generated by each MEC is too tremendous to transfer, and direct share the hash string to other operator base stations will also lead to privacy leaks. So a suitable performance aggregation method that can integrate the embedding in the MEC information aggregation is needed. A naive aggregate method is summed all content embedding in a single MEC, which is an effective method used in regular network embedding application in the graph representation. However, only dozens of content embeddings will be aggregated in a regular application, while a MEC may contain tens of thousands of contents. Simply add up all the MEC contents embedding directly, then the mixed embedding may only indicate the higher price content in the MEC while ignoring other content possible values not so high. Therefore, we introduce an effective MEC hybrid embedding method to balance low-value and high-value content.

We propose a graph embedding-based aggregate method to solve the MEC hybrid embedding problem and express our algorithm in Algorithm 1. Firstly, as described in [20], the demand of the content follows the Zipf distributions and we use it to simulate the MEC request sequence. Then, we use the used skip-gram algorithm [8] to maximize the co-occurrence probability so that the frequently visited content can be expressed as similar as possible, and at the same time, the representations between different contents are as different as possible. To let all content be represented, we add a sequence to represent all content in a single MEC to ensure that all MEC content can generate better embedding. The objective function of this step is the maximum the following probability:

$$P\left(w_o \mid w_c\right) = \frac{\exp\left(\boldsymbol{u}_o^{\top}\boldsymbol{v}_c\right)}{\sum_{i\in\mathcal{V}}\exp\left(\boldsymbol{u}_i^{\top}\boldsymbol{v}_c\right)} \tag{11}$$

After generating the content embedding, we can train the MECs embedding simultaneously, randomly initialize a MECs embedding, and then minimize the distance between the MECs embedding and the content embedding. Finally, we design the loss function for a single MEC to include embedding for a single content and the entire MEC content as Eq. (12). Then, SGD is used to optimize this function.

$$\begin{aligned} \text{minimize}_{\Phi} & -\log \Pr\left(\left\{v_{i-w},\cdots,v_{i+w}\right\}\backslash v_i \mid \Phi\left(v_i\right)\right) \\ & -\log \Pr\left(\left\{v_{i-w},\cdots,v_{i+w}\right\}\backslash MEC_i \mid \Phi\left(MEC_i\right)\right) \end{aligned} \tag{12}$$

Algorithm 1 Content Embedding Generate.

Input: Single MEC global content representation C
embedding size d
Zipf sequence length l
Output: content representation $C^{|V|\times d} = \{v_1, v_2, ..., v_n\}$
MEC representation $MEC^d = \{m_1, m_2, ..., m_n\}$
1: Initialization: MEC representaiton MEC^d
2: Generate sequence S use zipf distribution
3: Add a whole content sequence S^{whole} to S
4: **for each** $s = 0$ to $\mathcal{S}$ **do**
5: **for** each $s_j \in S$ **do**
6: **for** each $u_k \in S[j-w : j+w]$ **do**
7: $J(\Phi) = -\log \Pr(u_k \mid \Phi(v_j))$
$-\log \Pr(MEC \mid \Phi(v_j))$
8: $\Phi = \Phi - \alpha * \frac{\partial J}{\partial \Phi}$
9: **end for**
10: **end for**
11: **end for**

5.3 Bi-part network recommend and update

With the MEC representation obtain, each MEC of different operators can deliver its embedding to the base station. When the base station receives a content request from the user, the base station calculates the similarity between the user request and the MEC embedding, and then selects the MEC node with the highest similarity request content. When MEC receives the request, it finds whether the content is in its own storage. If it does not exist, it feeds back to the base station and sets a counter. When the failure ratio reaches a certain threshold or content in the MEC replacement reaches a certain percentage, it recalculates the MEC's embedding.

6 Blockchain design

The purpose of our design is to build a user-friendly MEC system. Suppose a customer belonged to operator A uses operator B's infrastructure services, which will be common in real life because of the nearby service principle. In that case, operator A &B should coordinate the cost allocation problem by themselves and let the user only pay his operator's cost as usual.

This scenario leaves out the issue of a complete and trusted record of each user's use of the cross-operator service. Suppose the operators are selfish; it is not feasible to let them keep their own records, as they may fabricate false usage records to gain profit. An independent third party is introduced to monitor and record all the use of cross-operator service in the region is a kind of solution, but like all other third-party centralized systems, the operators have to share part of their interests to others, and the third party may be attacked or has a single point failure. This is not the optimal solution in this scene.

Blockchain provides a solution for building distributed trusted records between untrusted peers. Blockchain is a distributed database that maintains a growing list of records (for example, account transfer records, the cross-carrier service giant in our scenario) stored in blocks and linked together by the previous block, which are immutable. For this reason, blockchain technology can build trusted records between untrusted peers, as all records in this chain are verified by nodes in the blockchain network that are not associated with the records and are permanently open for later queries. In the case of malicious activity, the validation node can track the recorded properties and resolve the problem.

6.1 Blockchain construction

As illustrated in Fig. 3, multiple MEC servers are belonging to multiple operators in a region, and in our design, each operator in each region has a MEC server who is rich in storage and computing resources as the node to deploy the blockchain—or miner as it is called in Bitcoin. This deployment is because we do not want to pressure the MEC servers for storage and calculation. After all, it is frequent for the miner node to perform transaction verification and save and synchronize ledger data. In the dense MEC servers in 5G environment, it seems unnecessary for each base station to store massive ledger data.

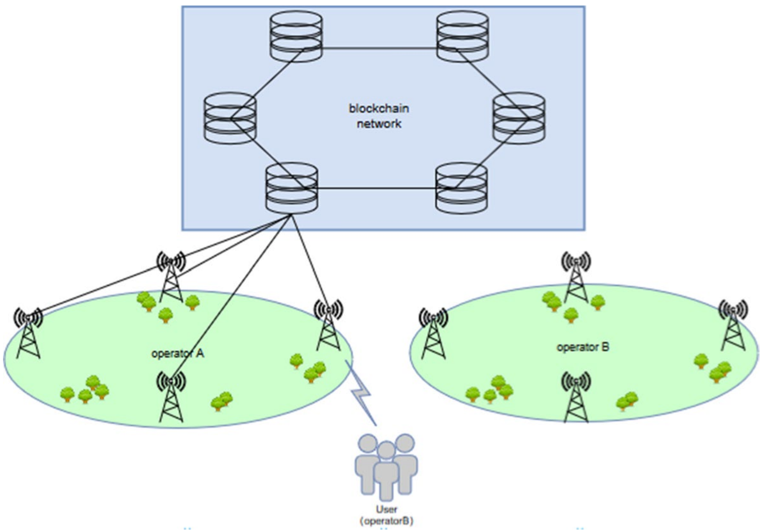


Fig. 3 blockchain scenario

So when the MEC server provides services across operator domains, it prepares a transaction to notify the miner's node of its operator in this region. A transaction is stipulated to include the following contents: time, area, service content, how many operators X should pay to operator Y, and the verification part. The miners will then consolidate transactions into a block and add new blocks to the blockchain under consensus rules.

6.2 Consensus and verification

Considering scenario and consensus efficiency issues, we decided to use the consortium blockchain, which is not a complete distrust between nodes. In fact, nodes belonging to the same operator have no reason to cheat on each other. Therefore, the consensus similar to Bitcoin POS that each block needs about 10 minutes of confirmation time for security seems unnecessary. Considering the number of cross-operator services, a consensus for shorter confirmation times under the consortium blockchain would be more appropriate.

Thus, we improved the DPOS consensus to accommodate business requirements; specifically, we added the part that verifies the transaction's validity. A DPOS consensus can be simply explained as follows: All nodes elect a group of block producers, then block producers produce blocks in a randomly given order, and the produced blocks can be finally determined (that is, irreversibly join the blockchain) after being signed and authenticated by more than two-thirds of the block producers.

In our system, this consensus process can be described as three phases:

(1) All nodes vote for 21 block producers proposed in [21], and the top 21 in the vote will automatically become the block producer of this round. Each node's voting right is related to each node's honesty. (The initial honesty of each node is the same.) If a node publishes a transaction without providing a service to the user, its honesty will be deducted if the verification fails.

(2) Each block producer generates blocks in a random order, and the block generation time is 2s. A block that is not generated in the producer's own build block time is invalid.

(3) When the block producer receives a new block, it needs to do two verifications. First, whether the corresponding producer has produced the block in a legal time; and second, whether the transaction contained in the block is real and valid, only both validations succeed, it signs the block. When an ordinary node receives a new block, it checks the signatures on the block, and only blocks with more than two-thirds of the producers' signatures (i.e., 15 +) are added to the ledger of the node.

Next, we will explain how the block producer verifies that the transaction is real and effective. When MEC servers provide a service to the user (such as providing their own cached resource), MEC servers will send a challenge to the user and receive a user response. Then, when a transaction is initiated, the MEC server will attach its own challenge and the answer and a signed response from the user. The block producer determines whether the user initiated the response and whether it met the answer. To be clear, if the response meets the answer, the user must have received the service provided by the MEC server. Such a challenge/response mechanism draws on the idea of POS.

A challenge-response mechanism can be described in detail below. MEC servers will regard file $\mathbf{f}$ as an n-dimensional vector and each element of it will be tagged. When serving the users, it will send a file not only $\mathbf{f}$ but also all tags $\mathbf{t}$. Meanwhile, it is required to send a challenge, which is a random vector $\mathbf{c}$ and then receive a response that should be a tag τ computed using $\mathbf{f}$,$\mathbf{t}$ and $\mathbf{c}$ along with user's public key. The verification node will see the result tag calculated by the MEC server itself and the τ, which is encrypted by the user's private key and received by the MEC server. All they need to do is to decrypt τ with the public key and compare it with the tag to see if equal.

7 Simulation results and analysis

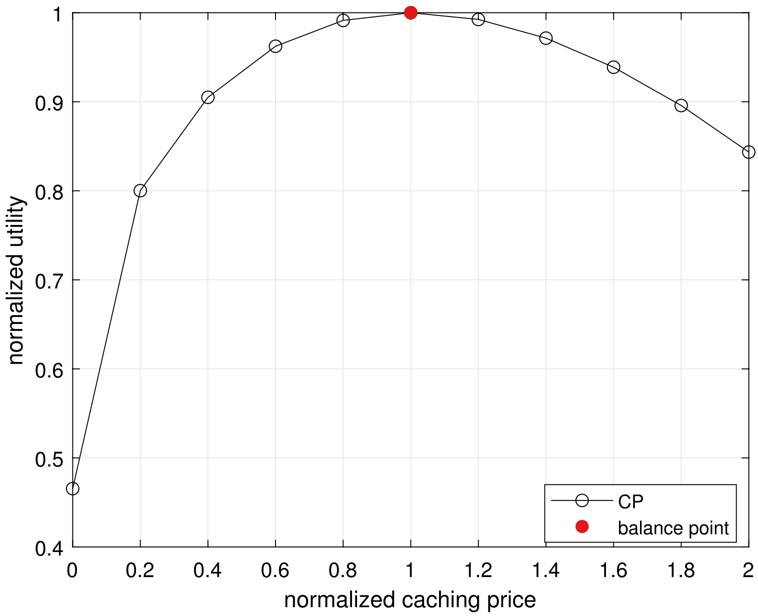
In this section, we present the simulation results and mainly evaluate the following: (1) the utilities of CP and FNs at Stackelberg equilibrium compared with at other action points; (2) the matching success rate use our graph embedding-based MEC framework compared with other strategies.

7.1 Experimental setup for cache scheme

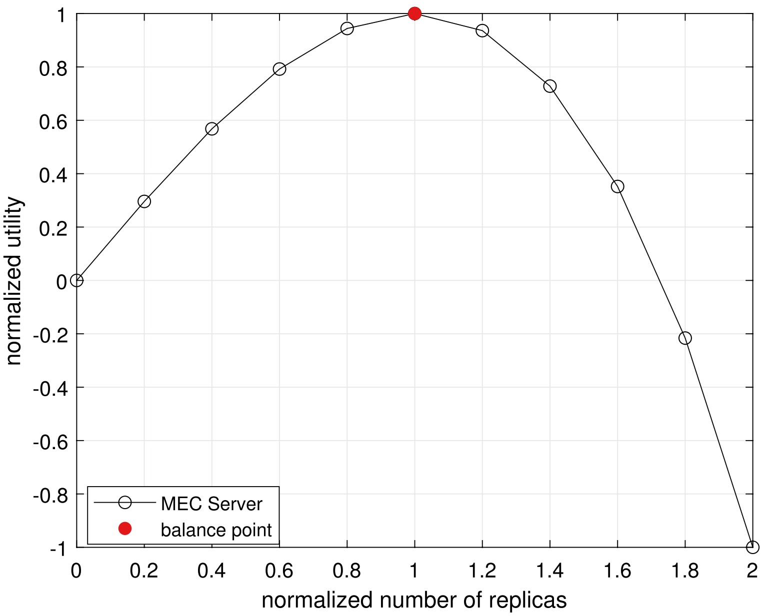
We consider a 5G edge computing scenario with 7 CPs, 7FNs, 100 users. The FNs are distributed in a cellular fashion with distance between two FNs as 150m. Users are randomly distributed and may be served by multiple FNs if it locates in the overlapping area of two FNs. Content is divided into 20 classes according to popularity. FNs provide caching services for CPs. The caching price and caching amount decision between CPs and FNs are made based on Stackelberg game-based caching scheme. Users made 2000 requests for content following Zipf distribution with parameter α. The users' requests and content source matching problem are solved by graph-based recommendation scheme.

7.2 Utilities of CPs and FNs at Stackelberg equilibrium

In Fig. 4, we demonstrate the joint optimization of utilities of CPs and FNs at Stackelberg equilibrium. Setting the content distribution parameter α as 1.6. We obtained the Stackelberg equilibrium using the method elaborated in Sect. 4 which are shown in Fig. 4 as filled red dot. Deviation from the Stackelberg equilibrium results in the decline of the utilities of CPs and FNs. Therefore, the simulation results proved the correctness of the Stackelberg equilibrium derived from mathematical analysis.



(a) Utility of CP



(b) Utility of FN

Fig. 4 the utilities of CP and FN at Stackelberg equilibrium

7.3 Experiment for graph embedding matching

We first simulated a virtual network scenario in MATLAB. There are five operators in this network environment, and then, each operator sets up 1000 users, and each operator deploys the same number of base stations and MEC servers. The simulated user requests the content in the MEC during use. The request sequence follows the Zipf distribution according to the content's popularity, and the Zipf parameters are the same as the above cache scheme experiment. We verify our proposed method's overall system delay performance and common methods under different MEC cache capacities and different MEC deployment quantities through experiments. We compare our graph-based method against the following traditional method:

(1) RandomSelect: The base station processes the request by randomly choose the MEC server.
(2) LocalIndex: The base station processes the request by search the index. However, these indexes only include their own MEC server. When the requested content cannot be found, the base station request content from the cloud server.
(3) GlobalIndex: Same as the LocalIndex method, which base station requests content use the index, but instead of partially content index, there exists a MEC's global index. It is an ideal situation but may cause privacy leakage.

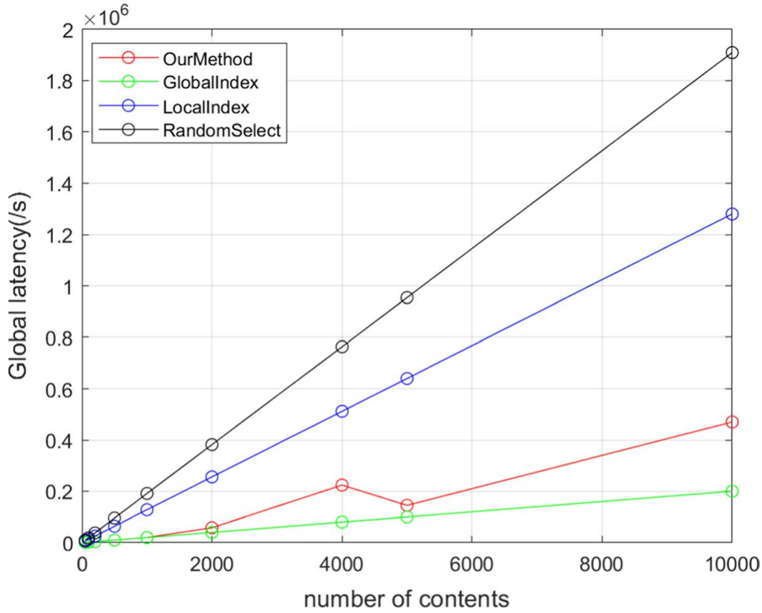
7.4 Impact of the MEC content capacity

In this experiment, we set the number in MEC to 100 to explore the influence of different MEC server storage capacity on our graph embedding model. Each base station user-generated 500 content requests in the experiment, and these requests are randomly allocated to different base stations. The experiment measured GBRM's performance and other methods in the overall network request delay performance of MEC with different cache storage capacities.

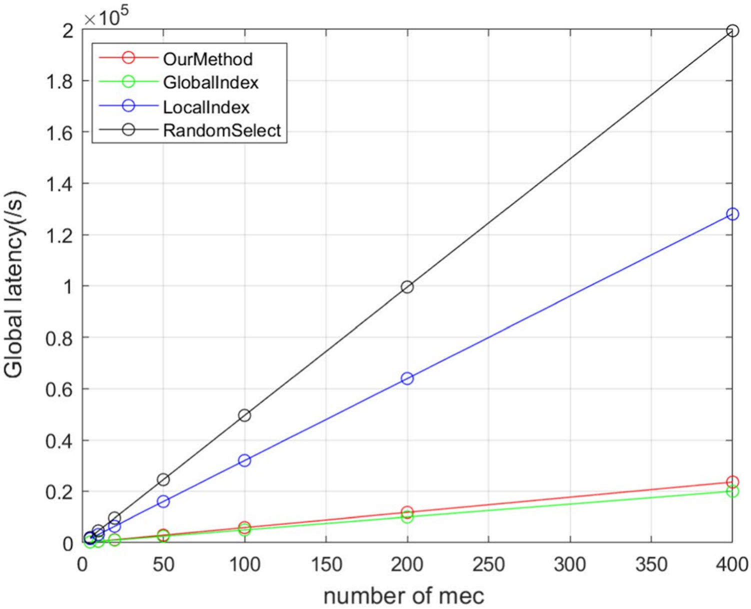
As we can see from Fig. 5a, GBRM is significantly better than the RandomSelect and the LocalIndex method in global latency and close to the ideal situation of the global index. After the MEC cache capacity expanded to 5000, the global latency showed some decline, and at 10,000, there was some increase, which may mean that our GBRM method can obtain better results in the capacity of 4000–5000. In general, GBRM can reach a good compromise between global indexing and random selection with MEC content privacy protection.

7.5 Impact of the MEC quantity

In this experiment, we set the cache capacity in the MEC to 4000 to explore the impact of different MEC deployments in the network on the overall network request delay performance. The remaining experimental settings are consistent with the above MEC content capacity experiment. From the experimental results Fig. 5b, we can find that the performance of GBRM is very close to the global indexing method



(a) MEC capacity effect



(b) MEC quantity influence

Fig. 5 The impact of MEC capacity and quantity on system delay performance and far superior to the local indexing and random selection methods in global latency. Therefore, through MEC capacity and quantity experiments, we can prove that the method based on graph embedding matching can more effectively help users match the required resources across different operators and simultaneously protect the content privacy of different operators.

8 Conclusion

In this paper, a multi-stakeholders 5G MEC system for content caching and distribution with graph embedding recommend system and blockchain assisted. We first studied the interaction of CPs and MEC server and modeled as a Stackelberg game. We then propose a graph embedding-based content recommended system to solve the user requests and content sources matching problems. Finally, to solve the multiple different MEC stakeholders, a blockchain-based settlement scheme is proposed. Our framework systematically solves the 5G mobile environment MEC caching problem in an elegant way. In the future, We hope to test our algorithm in actual 5g edge computing scenarios and find issues in actual deployment.

Acknowledgements This work is supported by the National Science Foundation of China (NSFC 62072012), Key-Area Research and Development Program of Guangdong Province (2020B0101090003), Shenzhen Project (JSGG20191129110603831), and Shenzhen Key Laboratory Project (ZDSYS201802051831427).

References

1. Abbas N, Zhang Y, Taherkordi A, Skeie T (2018) Mobile edge computing: a survey. IEEE Internet Things J 5(1):450–465
2. Li J, Chen H, Chen Y, Lin Z, Vucetic B, Hanzo L (2016) Pricing and resource allocation via game theory for a small-cell video caching system. IEEE J Sel Areas Commun 34(8):2115–2129
3. Lei K, Xie Y, Shi J, Zhang H, Zhang G, Bai B (2018) Optcaching: a Stackelberg game and belief propagation based caching scheme for joint utility optimization in fog computing. In: 2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS), pp 747–754
4. Zheng Z, Song L, Han Z, Li GY, Poor HV (2018) A Stackelberg game approach to proactive caching in large-scale mobile edge networks. IEEE Trans Wirel Commun 17(8):5198–5211
5. Xiong Z, Feng S, Niyato D, Wang P, Leshem A, Zhang Y (2018) Game theoretic analysis for joint sponsored and edge caching content service market. In: 2018 IEEE Global Communications Conference (GLOBECOM), pp 1–7
6. Sarwar B, Karypis G, Konstan J, Riedl J (2001) Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th International Conference on World Wide Web, pp 285–295
7. Perozzi B, Al-Rfou R, Skiena S (2014) Deepwalk: online learning of social representations. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 701–710
8. Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781
9. Tang J, Qu M, Wang M, Zhang M, Yan J, Mei Q (2015) Line: Large-scale information network embedding. In: Proceedings of the 24th International Conference on World Wide Web, pp 1067–1077
10. Dong Y, Chawla NV, Swami A (2017) Metapath2vec: scalable representation learning for heterogeneous networks. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 135–144

11. He Y, Song Y, Li J, Ji C, Peng J, Peng H (2019) Hetespaceywalk: a heterogeneous spacey random walk for heterogeneous information network embedding. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, pp 639–648
12. Gao M, Chen L, He X, Zhou A (2018) Bine: bipartite network embedding. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, pp 715–724
13. Huang W, Li Y, Fang Y, Fan J, Yang H (2020) Biane: bipartite attributed network embedding. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Ser. SIGIR '20. Association for Computing Machinery, New York, pp 149–158 [Online]. https://doi.org/10.1145/3397271.3401068
14. Nakamoto S (2018) Bitcoin: a peer-to-peer electronic cash system [online]. https://bitcoin.org/bitcoin.pdf
15. King S, Nadal S (2012) PPCoin: peer-to-peer crypto-currency with proof-of-stake
16. Vasin P (2014) Blackcoin's proof-of-stake protocol v2 [online]. https://www.blackcoin.co/blackcoin-posprotocolv2-whitepaper.pdf
17. Kamara S (2013) Proofs of storage: theory, constructions and applications. In: Muntean T, Poulakis D, Rolland R (eds) Algebraic Informatics. Springer, Berlin, Heidelberg, pp 7–8
18. Ateniese G, Kamara S, Katz J (2009) Proofs of storage from homomorphic identification protocols. In: Matsui M (ed) Advances in Cryptology—ASIACRYPT 2009. Springer, Berlin, Heidelberg, pp 319–333
19. Rivest R, Dusse S (1992) The MD5 message-digest algorithm, internet request for comments. Internet Request for Comments (RFC) 1321
20. Newman ME (2005) Power laws, pareto distributions and zipf's law. Contemp Phys 46(5):323–351
21. Elrom E (2019) EOS.IO wallets and smart contracts. Apress, Berkeley, pp 213–256 [Online]. https://doi.org/10.1007/978-1-4842-4847-8_6

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Kai Lei[1] · Hao Ye[1] · Junjie Fang[1] · Peiwu Chen[2] · Liangjie Zhang[3] · Jing Xiao[2]

Hao Ye
yehao@pku.edu.cn

Junjie Fang
1801213354@pku.edu.cn

Peiwu Chen
chenpeiwu194@pingan.com.cn

Liangjie Zhang
zhanglj@kingdee.com

Jing Xiao
xiaojing661@pingan.com.cn

1 ICNLAB, School of Electronic and Computer Engineering (SECE), Peking University, Shenzhen, China

2 Ping An Technology (Shenzhen) Co., Ltd., Shenzhen, China

3 Kingdee Software Company Limited, Shenzhen, China