EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

A collaborative service resource evaluation model based on trust network

Jiantao Shi[1], Likun Liu[1], Xiangzhan Yu[1] and Hui Lu[2*]

*Correspondence: luhui@gzhu.edu.cn
[2] Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou 510006, China
Full list of author information is available at the end of the article

Abstract

In the era of the 5G network, network traffic grows rapidly. Edge computing will face the challenges of high bandwidth, low latency, high reliability and other requirements of 5G network services. Due to the limited resources of node communication, computing and storage, under the sudden, intensive and high-traffic task request, edge computing will suffer from network jitter, excessive delay, access congestion, service failure and low distribution efficiency. In order to ensure network service quality and improve service efficiency, it is necessary to construct an effective collaborative service mechanism, motivate nodes to participate in cooperation, integrate network service resources and self-adapt to manage collaborative services. Therefore, establishing an effective node and service evaluation system to identify reliable resources and nodes is an effective way to improve the overall availability and reliable services of edge computing network. In this paper, we summarize and analyze the key technologies of the current collaborative service organization incentive and trust mechanism. This paper presents an attack-resistant node and service evaluation system based on a trust network. The system includes a voting collection mechanism, a trustable node selection mechanism, a questioning response mechanism and a punishment mechanism. The experiments prove that the system has a strong ability to resist attacks and is superior to the existing reputation evaluation model in terms of its performance. It can effectively improve the collaborative efficiency of edge computing and guarantee the quality of network service.

Keywords: Trust model, Edge computing, Traffic analysis, 5G network

1 Introduction

The amount of equipment and the data it generates in the 5G era is growing rapidly. Streaming application service for large traffic and large connection has become the main service of communication network traffic. By 2020, the total amount of global data will exceed 40ZB [1], among which 45% of IOT data will be processed at the edge of the network. According to the Cisco Data Vision Network Index (VNI: Visual Networking Index) [2], starting in 2016, more traffic is being unloaded from cellular to Wi-Fi. By 2021, more than 78 percent of global mobile data traffic will be video, IP video traffic will account for 82% of the entire network flow, video, gaming and multimedia will occupy the entire flow of more than 85%, the content delivery network (CDN) traffic will account for 71% of the entire network flow [3]. In the face of such massive data access

[4], when implementing large-scale network data communication based on the traditional computing model with centralized big data computing and storage as the core, it is inevitable that the centralized big data processing capacity cannot meet the massive edge data [5]. Research shows that edge computing with cloud computing as the core is one of the effective methods to solve this problem [6, 7]. Edge computing increases the processing capacity of performing task computing and data analysis on network edge devices, transfers some or all computing tasks of the original cloud computing model to network edge devices, reduces the computing loads of cloud computing centers, reduces the pressure of network bandwidth and improves the data processing efficiency [8]. The key to optimizing edge computing collaborative services lies in how to select reliable collaborative service allies, and how to allocate, schedule and migrate collaborative tasks after collaborative services are organized [9]. In order to solve the problems of node reputation and collaborative service quality evaluation under the conditions of unstable communication link-state, limited node communication resources and unpredictable node behavior, a credibility evaluation model under an edge computing network is proposed. The method constructs the underlying implicit trust relationship based on the identity, state and behavior information of collaborative nodes in the network. And depending on the type of the task, user satisfaction was modeled based on service quality evaluation such as delay, jitter, packet loss and expected rate. A QoS evaluation feedback of collaborative service quality based on credibility is realized. The experimental results show that this mechanism can reflect the trust relationship between nodes under different link states and interaction results more effectively and promote the cooperation of nodes more effectively.

2 Related works

The evaluation of service credibility is the main way to solve the problem of node selection in collaborative networks, but the authenticity of the evaluation is also the key security problem that troubles the evaluation system [10]. To address this problem, many researchers have proposed some reputation and trust evaluation mechanisms [11, 12], including the reputation mechanism based on the node subject itself, which is representative: EigenTrust proposed by Kamvar et al. [13] and PeerTrust proposed by Xiong et al. [14]. EigenTrust is a reputation mechanism based on global information. By virtue of the transitivity of trust, when the trust value is calculated, the global trust value of the node itself is taken as the recommended weight. It is believed that there is a correlation between service performance and the credibility of recommendation information, that is, the information recommended by nodes with good service is more reliable, while the evaluation feedback provided by nodes with poor service is also more likely to be inaccurate. However, if the attacker adopts a more optimized attack mode, this assumption will help the attacker to improve the success rate of pollution and denigration attacks [15, 16]. The security of this method is completely dependent on the pre-existing central server nodes in the network, which is difficult to achieve in the real edge computing network environment. Dou et al. [17] have improved EigenTrust's iterative convergence and safety, but the improved model still has problems of efficiency and safety. Safety of EigenTrust cannot be improved by punishment alone. PeerTrust is a credibility mechanism based on local information, which comprehensively considers multiple factors affecting

trust value, including transaction evaluation, transaction times of nodes, credibility of feedback source, transaction context and so on. The pure distributed trust value calculation method is used to calculate the trust value based on the principal similarity, that is, to give higher recommendation weight to those who have proposed similar evaluation to frequent transaction nodes, and to improve the authenticity of transactions through secure submission mechanism and transmission feedback mechanism. But, by adopting the same subject credibility assumption as EigenTrust, PeerTrust is also vulnerable to complex attacks. The above models all adopt the credibility mechanism based on global information with the help of the transmissibility of trust. However, if the attacker adopts more optimized attack methods, such as Sybil attack or conspiracy attack, this hypothesis will help the attacker to improve the success rate of pollution and slander attacks. The security of such method is completely dependent on the existing central server nodes with high reputation in the network, which is difficult to realize in the actual distributed edge network environment.

In order to solve these problems, some models based on social relations have emerged [18–20]. The guidance graph model proposed by Danezis et al. uses social networks to propose correctness standards for secure routing [21] and gives preliminary suggestions on concrete implementation. Yu et al. successively proposed the SybilGuard [22] and SybilLimit [23] systems, which make use of the feature of fast mixing of social networks, to help, as the mixing is in the distributed network, with arbitrary benign nodes communicating with another honest node with high probability. After this, Lesniewski-Laas [24] used social networks to propose a DHT system that could resist 'witch attacks.' All of these applications don't really solve the problem of resource evaluation in a distributed network, only SumUp system [25] uses social networks to evaluate objects. But SumUp relies on a trusted server to store historical information about node transactions, rather than being applied to a purely distributed network [26]. In addition, SumUp defines friendship as bi-directional. That is, if A is a friend of B, B must be a friend of A, which is not exactly similar to the trust relationship between people in practice.

In this paper, we also propose a trusted resource evaluation system in edge computing collaborative service network by means of social trust network and describe the trust relationship between nodes in a one-way way. It can effectively evaluate resources, resist most cunning attack strategies and perform better than existing object-based trust mechanism models.

3 Methods

3.1 System overview

As an independent object evaluation model, the evaluation system has nothing to do with the specific structure of distributed network. As shown in Fig. 1, all nodes, including honest users and adversaries, provide collaborative services. An honest user collaborates with the network to publish and retrieve service content of interest and always ensures that the exact service content is provided. Rival nodes try to offer fake services and increase their popularity. Without a special content rating and evaluation system, users cannot accurately distinguish between true results and false results. Our reputation evaluation system uses network user group to score the used services, and any user in the system has a unique identity to put forward his

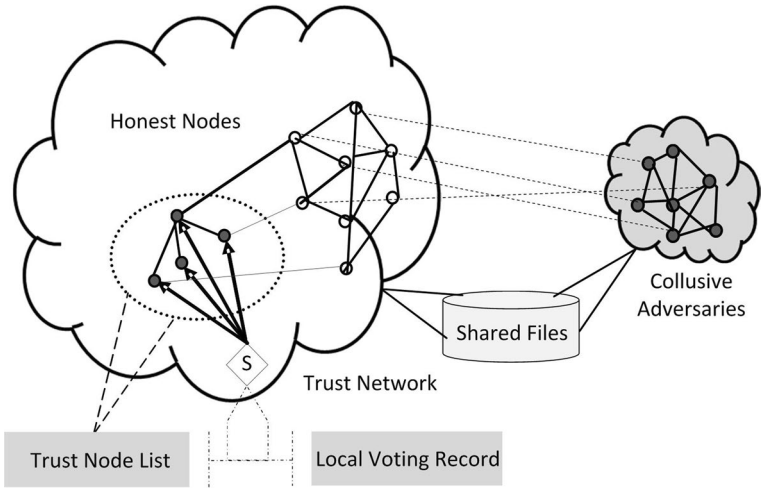

Fig. 1 Network model of the service rating system. As an independent object evaluation model, the evaluation system has nothing to do with the specific structure of the distributed network. As shown in figure, all nodes, including honest users and adversaries, provide collaborative services. Honest users cooperate with the network to publish and retrieve the service content of interest and always ensure that accurate service content is provided. Competitor nodes try to provide fake services and improve their visibility. Without a special content rating and evaluation system, users cannot accurately distinguish between true and false results. Our reputation evaluation system uses the network user group to score the services used. Any user in the system has a unique identity and can make his own evaluation on the services in the system

own evaluation of the services in the system. The unique identity in the system cannot be counterfeited by some existing ID generation technology and access control technology [27].

Object evaluation value R(k,i) represents user i's evaluation of the authenticity of service k, value 1 represents the service description is consistent with the service content, and value 0 represents that the service description is false. We will create a trust network independent of the overlaying network. On the one hand, one node can choose the trust nodes according to their own experience and social cognition and establish a one-way trust relationship with them. On the other hand, high-quality nodes can be selected to join their own trust node list by evaluating the service quality.

In order to ensure the resource discrimination and efficiency of the Trust network, each node will store two important data structures locally: the Trust Node List (TNL) and the Local Voting Record (LVR). Nodes in TNL can be added manually, and it will only send and forward query messages to nodes in TNL. The types of messages used include: votes Query message, voting records exchange message, votes challenge Request, etc. LVR adopts a second-level data structure as shown in Fig. 2. Hash the service IDS according to the size of the hash table as the first-level hash and then establish the second-level index according to the service IDS after the data items are read. The left side is the hash of the confirmed evaluation, and the right side is the hash of the unconfirmed evaluation. Unevaluated hash retains the ID of the evaluated node according to the evaluated value.

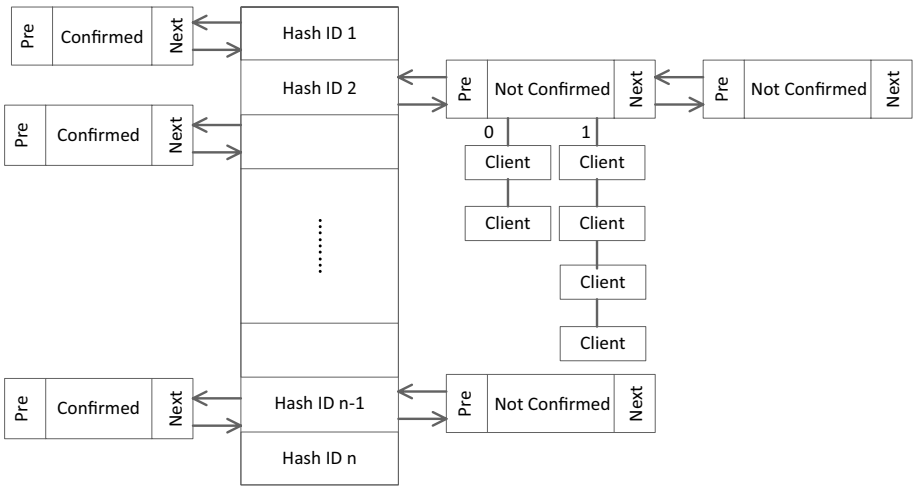

Fig. 2 Two level hash table of LVR. LVR adopts a second-level data structure. Hash the service IDS according to the size of the hash table as the first-level hash and then establish the second-level index according to the service IDS after the data items are read. The left side is the hash of the confirmed evaluation, and the right side is the hash of the unconfirmed evaluation. Unevaluated hash retains the ID of the evaluated node according to the evaluated value

3.2 Evaluation collection mechanism

Before starting this mechanism, the user should have formed his own list of trust nodes

Algorithm 1: v.VotesQueryingRequest(Λ, m)

Input： **Λ – services to be evaluated; m – number of trusted nodes**

Output： **V(Λ) – evaluation score of services；**

1. $V(\Lambda) \leftarrow \Phi$
2. **for** $i \leftarrow 1$ **to** m **do**
3. $u \leftarrow TrustedPeerChoice(v.TNL)$
4. $V_i(\Lambda) \leftarrow v.Query(\Lambda, u)$
5. **endfor**
6. $V(\Lambda) \leftarrow v.SumUp(V_1, \ldots, V_m)$
7. **return** $V(\Lambda)$

and selected the set of services to be evaluated through the service search process.

The algorithms used in the evaluation collection process include Algorithms 1 and 2. Algorithm 1 represents the process of collection request initiated by node V. V selects m trust nodes U through the node selection mechanism, issues concurrent query request and then processes the final evaluation set according to the returned results of the trust network. If the evaluation of a service is returned from multiple nodes, the final evaluation value is calculated by formula (1).

$$v.R_k \left\lfloor 2\left(\sum_{u=1}^{m}\left(v.C_u \cdot R_u^k\right) \Big/ \sum_{u=1}^{m} v.C_u\right)\right\rfloor \tag{1}$$

where $v.C_u$ represents a reputation evaluation of node V to node U, which is stored locally to node V. It is a decimal value between [0,1]. Not only does the node have a rating value for the trust node, but also has an initial rating value of 0.5 for the untrust node. This calculation method not only ensures that the final value tends to the majority of the evaluation value in the result, but also prevents the fraud of the node with low

reputation. Node V will save the evaluation given node and evaluation recommended node in unconfirmed LVR. If the request message has a previous hop node, then randomly select a node from evaluation nodes whose evaluation value is equal to the final calculated value. Then returned the node to the previous hop.

Algorithm 2: u.QueryingProcess(Λ, v)

Input： Λ – services to be evaluated; m –Source node of the query

Output： u.V(Λ) – evaluation score of services given by U ;

1. $u.V(\Lambda) \leftarrow \Phi$
2. $(\Lambda_{LVR}, u.V(\Lambda_{LVR})) \leftarrow u.Init(u.LVR)$
3. *If* $\Lambda\text{-}\Lambda_{LVR} \neq \Phi$ *then*
4. $u.V(\Lambda\text{-}\Lambda_{LVR}) \leftarrow u.VotesQueryingRequest(\Lambda\text{-}\Lambda_{LVR}, r)$
5. $u.V(\Lambda) \leftarrow \{u.V(\Lambda_{LVR}), u.V(\Lambda\text{-}\Lambda_{LVR})\};$
6. $u.VotesQueryingResponse(u.V(\Lambda), v);/$
8. **return** $u.V(\Lambda)$

Algorithm 2 represents the processing flow of nodes receiving evaluation requests. In the specific implementation, the evaluation value will be returned in batch packaging to ensure system efficiency.

3.3 Evaluation feedback mechanism

In order to punish the wrong evaluation node, an evaluation feedback process is also provided in the evaluation collection process. The feedback initiator will judge the authenticity of the service according to the received service, and if a bad service is obtained, it means that the evaluation provider is a malicious node. In order to ensure the identification of the node identity and the identification of the message signature, the node ID is calculated by the public key of the node. If the identity of the evaluation provider is wrong, it means that the evaluation sponsor has cheated, and the referee will be punished through the punishment mechanism. On the contrary, if the identity of the evaluation provider is correct, the questioning mechanism will be activated to judge its

Algorithm 3: v.Feedback(V(f))

Input： V(f) – false evaluation of service f

1. *(u,w)←v.VoteTrace(f,v.LVR)* // Get score given node and recommend node
2. *valid←v.varify(u,v,f);*//Authenticity verification
3. **if** *valid* **then**
4. *v.SendFeedback(u,v(f),w);* // Send feedback
5. $v.C_w \leftarrow v.challenge(w,f);$ // Send challenge request and adjust w's integrity value
6. **else**
7. *v.punish(u);*
8. **endif;**
9. *v.TNL←v.LowerTrustWight(u);* // Adjusts weight of U in the trust node list
10. **return;**

malice degree and adjust its local credibility value according to the results.

After receiving the evaluation feedback, the node should also judge the false feedback according to the situation. If the service evaluation is in the local confirmed evaluation record, it means that the feedback is false feedback and the feedback sender will be questioned. If it is in the non-confirmation record, feedback will only be forwarded back, and the signature of the feedback initiator shall be carried in the feedback message, which

prevents malicious users from damaging the evaluation capability of the system through

Algorithm 4: u.FeedbackProcess(V(f), v)

Input： V(f) – false evaluation of service f v – evaluating node

```
1. if u.Confirmed(V(f)) then
2.   u.Cv←u.challenge(v,f);
3. else
4.   u.Feedback(V(f));
5. return;
```

malicious feedback.

3.4 Evaluation challenge mechanism

When a node in the trust network discovers that the query is getting a false service, or that the query message from different paths is inconsistent, the evaluation query mechanism will be triggered. The specific process is shown in Fig. 3. To challenge node U, Node V first asks k friend nodes or high-trust nodes for several confirmed File ids and evaluations based on relevant keywords and File Collection request. After receiving the request, the friend node will select files related to the requirement keyword from its own LVR confirmed Hash store to form a file collection response and return it to the originating node. The originating node tries to select the file ids that have an intersection from the different node response messages, to prevent spoofing by malicious nodes. Node V will select the good service ID to constitute the "Votes Challenge Request" message of evaluation and collection. The whole process is secret to U. When U receives the message, it does not distinguish whether the message is a query request or a true evaluation collection request. If the node is a normal node, it will also start the processing flow of Algorithm 2 for evaluation collection, and even forward requests to its trust nodes. If

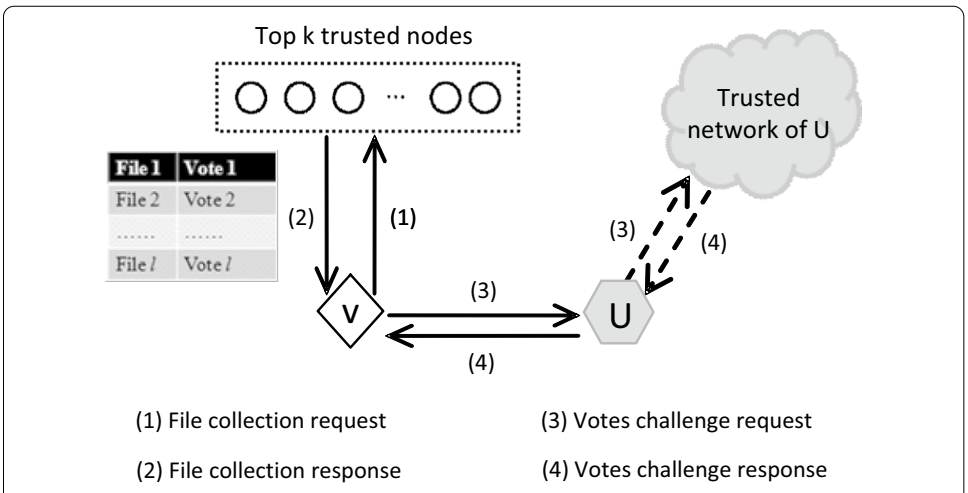

Fig. 3 Processes of the challenge mechanism. To challenge node U, Node V first asks k friend nodes or high-trust nodes for several confirmed File ids and evaluations. After receiving the request, the friend node will select files related to the requirement keyword from its own LVR confirmed Hash store to form a file collection response and return it to the originating node. The originating node tries to select the file ids that have an intersection from the different node response messages, to prevent spoofing by malicious nodes. Node V will select the good service ID to constitute the "Votes Challenge Request" message of evaluation and collection. The whole process is secret to U

the node is a malicious node, it will also give a false evaluation information. Node V will calculate the questioning accuracy rate according to formula (2) after receiving the Votes Challenge Response.

$$\theta = \frac{|\Lambda_{\text{corrected}}|}{|\Lambda|} \tag{2}$$

where $\Lambda_{\text{corrected}}$ refers to the number of files obtained by the evaluation collection algorithm in the feedback message within the specified time that are the same as the previously saved known evaluation results, excluding the error evaluation and unknown evaluation. Value θ can be set and modified by users. In the basic setting of the evaluation system, we believe that if $\theta < 0.7$ means that the node U is not trusted or has low credibility. For nodes with low doubt accuracy, we will trigger the punishment mechanism by lowering their integrity value. Therefore, in order to better integrate into the trust network, malicious nodes will try their best to give correct evaluation to the evaluation request, which greatly weakens their attack capability.

3.5 Punishment and incentive mechanism

The purpose of designing punishment and incentive mechanism is to encourage honesty node to evaluate the service accurately after the service is completed. Before providing upload, node V will refer to the credit value $v.C_u$ of node U. If the credit value is lower than the threshold of the system, the request will be rejected. When multiple requests from other nodes are received at the same time, the upload order will be determined according to the order of good faith value, and the nodes with high good faith value will be selected first to provide services.

The value of node integrity will be combined with the local calculated value and the network recommended value. The local calculation value includes two parts. One is SP, the service quality point of the evaluated node, the specific calculation method is shown in formula (3); the other is MP, the malicious behavior point of the evaluated node, it includes direct malicious behavior and indirect malicious behavior, and the specific calculation method is shown in Formula (4).

$$v.SP_u = v.TR_u + v.RR_u \tag{3}$$

$$v.MP_u = \begin{cases} v.FR_u + (v.MR_u)^2, & v.FR_u \le FR_{\text{base}} \\ v.FR_u + (v.FR_u - FR_{\text{base}} + v.MR_u)^2, & v.FR_u > FR_{\text{base}} \end{cases} \tag{4}$$

The local credibility value LC is obtained through SP and MP, the calculation method is shown in formula (5), which satisfies $0 \le LC \le 1$.

$$v.LC_u = \frac{v.SP_u}{v.SP_u + \delta \cdot v.MP_u} \tag{5}$$

where $\delta > 1$ is the amplification factor of malicious behavior. When there is no false evaluation and recommendation, the local credibility of the node is calculated as 1. When false evaluation is provided directly or indirectly, the local credibility value will be less than 1. Nodes can repair integrity by providing correct evaluation. However, due to the introduction of malicious behavior amplification factor, when the node continues

to provide malicious evaluation, the speed of integrity value repair will lag behind the speed of integrity value attenuation. The nodes providing recommendation are only those have established trust pairs between TNL and local nodes. Because each trust pair node will have a similarity value, its calculation method is as shown in formula (6).

$$v.NC_u = \frac{\sum_{i \in v.TPS} (v.Sim_i \cdot i.C_u)}{\sum_{i \in v.TPS} v.Sim_i} \tag{6}$$

Value $v.TPS$ represents the trust pair set of node v. The recommended network trust value of node U obtained by node V is calculated by averaging the recommendation values of all trust pairs. This means that node V is more likely to trust and cooperate with nodes that have had a long-term trust relationship, because such nodes are more likely to obtain a high degree of similarity. By combining the local calculated value of trust and the recommended value of network, the trust value of node V to node U is obtained by formula (7).

$$v.C_u = \frac{\eta \cdot v.LC_u + v.NC_u}{1 + \eta} \tag{7}$$

the network node u recommendation trust value obtained is through to all trust recommended values for the mean similarity of nodes, node v said more believe that long time and he has the trust and cooperation of nodes, because the node can obtain high similarity. By combining the local calculated value of trust and the recommended value of network, the trust value of node V to node U is obtained by formula (7), where $\eta > 1$ is the amplification factor of local calculation, indicating that the node trusts the local calculation value more, while the network recommendation value is only used as a reference. When evaluating a new node, due to the lack of local calculation value, network recommendation value will be taken as the main judgment standard.

3.6 Evaluation source and feedback source validation

Malicious attackers will make use of the fraud of evaluation source and feedback source, and defame and slander other nodes, which will reduce the credibility of benign nodes, affect the probability of benign nodes being selected in the trust node selection mechanism, and even seriously affect the benign users' use of collaborative network for service. In order to guarantee the authenticity of evaluation and feedback, this paper presents a verification protocol for evaluation source and feedback source. The protocol is simple to implement and does not require the participation of any third-party nodes. You only need to set up the private key encryption mechanism and the certificate issuing mechanism for the message between the nodes. As the intermediary point of the intermediate node, you only need to save the list information of customers. Firstly, the attack behaviors targeted by the authentication mechanism include: (1) malicious users forge the identities of other nodes, provide service evaluation information and send it to query users; (2) the malicious node sends false feedback messages through the feedback message sending chain. Therefore, the system needs to ensure the following rules: (1) in the LVR of any node, the evaluation providing source node in the non-deterministic evaluation hash store cannot be falsified; (2) the feedback source node in the evaluation

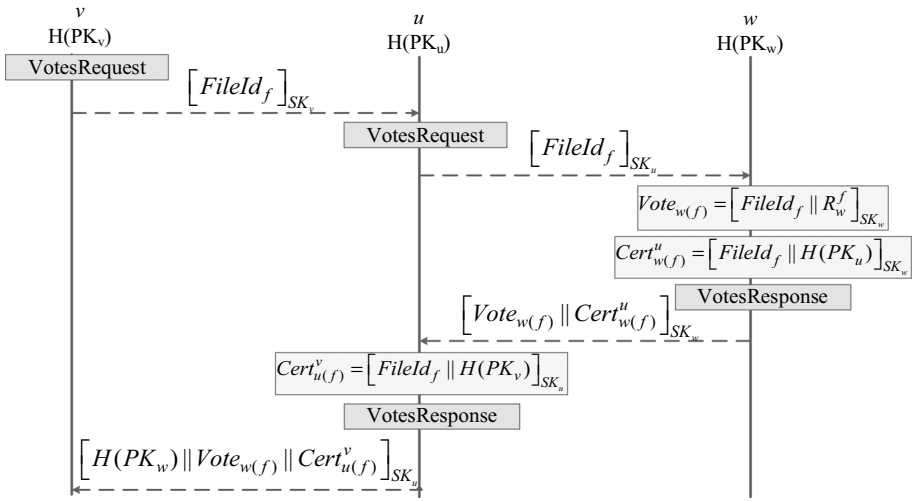

Fig. 4 Security protocol of evaluation collection process. Node W's evaluation ticket of service F, including the ID of service F and W's evaluation of f. The message is encrypted with the private key of W, the evaluation ticket will be returned along the query path, and other nodes cannot forge it. The other part of the body of the response message is the proof of transaction between the nodes

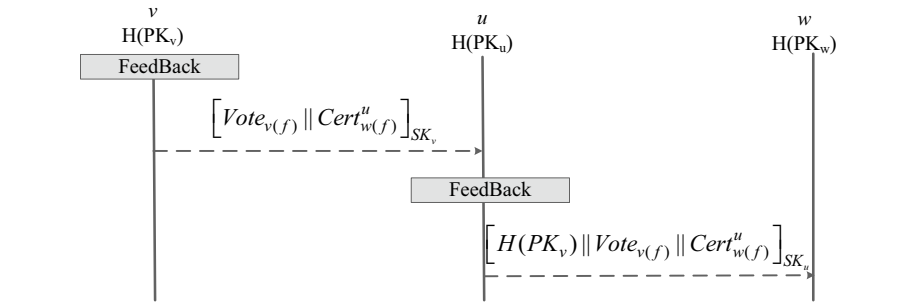

Fig. 5 Security protocol of evaluation feedback process. Node V carries the proof feedback source's evaluation ticket to service F, and the proof transaction certificate of node U and node V. When node U continues to feedback to node W, the steps are similar to the evaluation collection, but this time it carries the evaluation ticket from V. The protocol guarantees that the identity of any process node will not be counterfeited and that no node will deny its own behavior

feedback message cannot be falsified; (3) the feedback is sent back through the historical query chain, so the transaction information of the historical query cannot be falsified.

Figures 4 and 5 are the abstract protocol flow of the evaluation collection process and the evaluation feedback process. Message delivery is transmitted through cipher text, and the encryption key is the private key of the message-sending node. The evaluation collection request message includes the initial request and the forward request, which carries the ID or ID combination of the requested services. If the node receiving the request keeps the evaluation result of the node to be evaluated in the determined LVR, then the evaluation ticket is generated with the node's own private key as part of the response message body. In Fig. 4, node W's evaluation ticket of service F, including the ID of service F and W's evaluation of f. The message is encrypted with the private key of W, the evaluation ticket will be returned along the query path, and other nodes cannot

forge it. The other part of the body of the response message is the proof of transaction between the nodes. For example, W's proof of transaction to U includes the ID of node U and service F, and it is encrypted with the private key of node W to prove that node W has provided evaluation on F to node U, which is non-repudiation evidence. As the forwarding node, node U will generate its own transaction proof after receiving the response from W, and reply to node V with the evaluation ticket of W and the ID of W as the message body. Since the ID of the node is associated with its public key, node V can prove the identity of the service evaluation provider W by doing a hash function for the public key of node W and decrypting the ticket. If node V finds that the evaluation of service F recommended by U and provided by W is wrong, V will delay the query path to send the feedback message of service F evaluation. In the body of the message, node V carries the proof feedback source's evaluation ticket to service F and the proof transaction certificate of node U and node V. When node U continues to feedback to node W, the steps are similar to the evaluation collection, but this time it carries the evaluation ticket from V. The protocol guarantees that the identity of any process node will not be counterfeited and that no node will deny its own behavior.

4 Experiment

This paper evaluates the system's evaluation collection efficiency and its ability to resist malicious attacks through simulation experiments in different scenarios, including the comparison experiment with the existing object-based credibility mechanism Credence [28]. The simulation platform of the experiment adopted OverSim, in which the underlay of the underlying network was set as "Simple Underlay," and the number of integrity nodes in the simulation network was 5000. Since the purpose of the experiment was only to verify the delivery results of the real service in the network and the network's ability to resist the false service, the service uplink and downlink delays were ignored. According to the extended service evaluation model in this paper, all nodes form a large trust network through local TNL. The trust relationship and friendly relationship between nodes are set based on the actual dataset collected by Facebook and configured according to the experimental purpose. After joining the network, all honest nodes will provide real service and real evaluation, while malicious nodes that increase the experimental demand will provide false service and false evaluation.

5 Results and discussion

5.1 Ability to resist false evaluation

To verify the ability of the system to resist false evaluations, we compared our FRep system and Credence models through simulation experiments. In the experiment, 500, 1000 and 1500 false evaluation nodes were introduced, respectively, to provide false evaluation for all services. Among them, the number of collaborative services in the network is 10,000. LVR of each node has 2–3 random determination evaluations at the beginning of the simulation. Each simulation experiment runs 50 simulation rounds. According to Fig. 6, we can find that at a stable moment, the number of malicious nodes does not have a great impact on FRep and Credence. This is because even though the number of malicious nodes has multiplied, honest nodes still choose evaluation nodes through the trust network in a large proportion, and find malicious nodes through the questioning

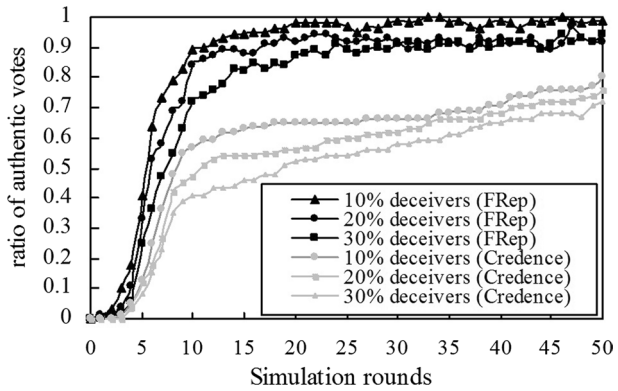

Fig. 6 Security protocol of evaluation feedback process. To verify the ability of the system to resist false evaluations, we compared our FRep system and Credence models through simulation experiments. 500, 1000 and 1500 false evaluation nodes were introduced, respectively, to provide false evaluation for all services. Among them, the number of collaborative services in the network is 10,000. LVR of each node has 2–3 random determination evaluations at the beginning of the simulation. Each simulation experiment runs 50 simulation rounds

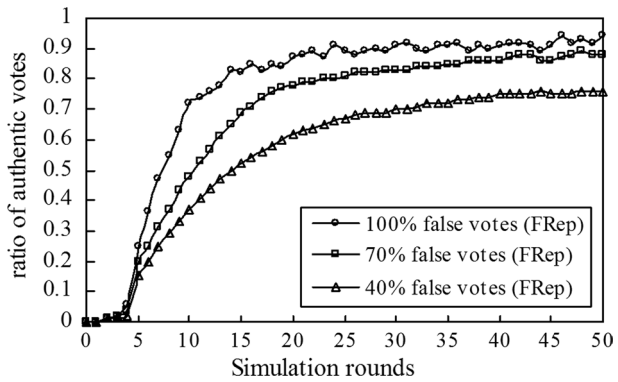

Fig. 7 Simulation result of system with tricky deceivers. Figure shows the ability of FRep to resist tricky deception obtained by experiments under the environment of 30% malicious nodes, in which the proportion of false evaluation provided by malicious nodes is 40%, 70% and 100%, respectively

mechanism. Therefore, FRep performance is not affected by the increase in malicious nodes. Experiments show that FRep performs significantly better than Credence.

5.2 Ability to resist tricky deception

Tricky deception node does not provide false evaluation for all services, but only returns false evaluation for a certain proportion of evaluation requests, and the rest of the evaluation returns true evaluation, so as to obtain a certain degree of trust. Figure 7 shows the ability of FRep to resist tricky deception obtained by experiments under the environment of 30% malicious nodes, in which the proportion of false evaluation provided by malicious nodes is 40%, 70% and 100%, respectively. When the proportion of false evaluation is 100%, the accuracy of the evaluation collection can quickly reach more than 0.8 within 10 simulation cycles. Even if the tricky deceptive nodes are adopted, the

proportion of false evaluation is controlled at about 40%, and the evaluation collection accuracy can reach above 0.7 after about 25 simulation cycles.

5.3 Ability to resist false evaluation

Figure 8 shows the evaluation collection capability of a collaborative service network under a collusion attack. The collusion node not only provides a false evaluation to the service, but also increases the credibility of the collusion node and lowers the credibility of the honesty node through malicious slander and flattery. The results showed that in terms of the ability to resist conspiracy to deceive, FRep performs significantly better than Credence, this is because FRep's evaluation and questioning mechanism limit the attack capability of collusion nodes. Positive nodes can form a tightly connected community through trust based on social attributes. The collusion nodes can be quickly identified through the cooperation between nodes, so as to cut off the attacking edge.

5.4 Influence of trust node number

A trust network based on community relationships plays an important role in FRep performance. Through simulation experiments, we analyzed the influence of network connectivity, that is, the number of online friends of each node on FRep performance. The experimental environment is still the collusion attack scenario. The collusion nodes account for 10% of the total number of nodes, and the number of friends is set as 14,10,6 and 2, respectively. From Fig. 9, we can see, when each node has only about 2 friends, it has a great impact on FRep performance, and the evaluation accuracy is not more than 50%. When the number of friends increased to 6, the evaluation accuracy can be significantly increased to 70%. When the number of friends increased to 10 or 14, the evaluation accuracy is increased to 90%, but the extent of increase becomes not obvious. It can be seen that FRep performance requires a small number of friends per node, but does not require a high number.

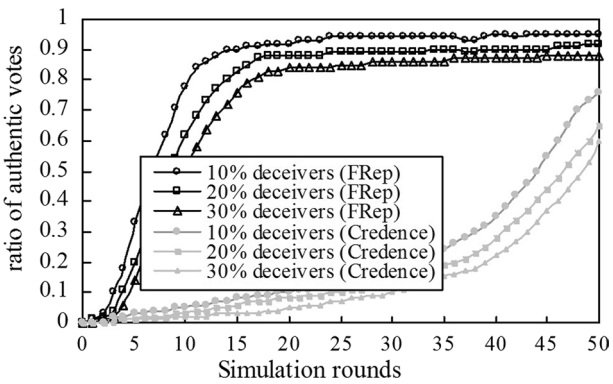

Fig. 8 Simulation result of system with collusive deceivers. Figure shows the evaluation collection capability of a collaborative service network under a collusion attack. The collusion node not only provides a false evaluation to the service, but also increases the credibility of the collusion node and lowers the credibility of the honesty node through malicious slander and flattery. We compared our FRep system and Credence models. The proportion of deceivers are 10%, 20% and 30%, respectively

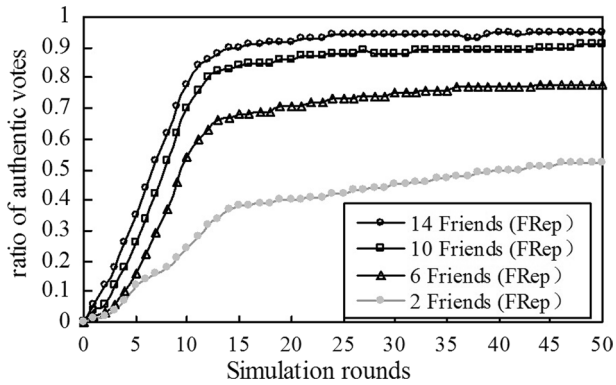

Fig. 9 Simulation result of different friends number. A trust network based on community relationships plays an important role in FRep performance. Through simulation experiments, we analyzed the influence of network connectivity, that is, the number of online friends of each node on FRep performance. The experimental environment is still the collusion attack scenario. The collusion nodes account for 10% of the total number of nodes, and the number of friends is set as 14, 10, 6 and 2, respectively

6 Conclusions

In the 5G network era, data-intensive applications oriented toward large traffic and large connections will become the main business. When collaborative service technology is applied to edge computing, it also brings new problems of information security. The FRep reputation mechanism proposed in this paper can effectively identify the behavior characteristics of nodes. The evaluation collection mechanism, node selection mechanism, evaluation questioning mechanism, incentive and punishment mechanism and historical evaluation exchange mechanism of the FRep system enable this model to resist large-scale cheating attacks. These attacks include witch attack, conspiracy attack and even the more subtle evaluation of deception attack. The simulation experiment shows that the FRep system has a better ability to resist attacks than the existing reputation model, and the collection process of object evaluation has very good efficiency.

Abbreviations

5G: Fifth generation; IoT: Internet of things; CDN: Content delivery network; DHT: Distributed hash table.

Acknowledgements

The authors thank the anonymous reviewers and editors for their efforts in valuable comments and suggestions.

Authors' contributions

System design, JS and LL; Methodology, JS and XY; Software, JS and HL; Validation, LL and XY; Writing—original draft, JS and HL; Writing—review and editing, JS. All authors read and approved the final manuscript.

Funding

This work is supported by National Natural Science Foundation of China (61872111, 61402137, 61872111) and National Key R&D Program of China (2016QY05X1000).

Availability of data and materials

Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

Declarations

Competing interest

The authors declare that they have no competing interests.

Author details

[1] School of Cyberspace Science, Harbin Institute of Technology, Harbin 150001, China. [2] Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou 510006, China.

Received: 13 January 2021 Accepted: 2 March 2021
Published online: 09 March 2021

References

1. V. Turner, J.F. Gantz, D. Reinsel, et al, The digital universe of opportunities: rich data and the increasing value of the Internet of things. https://www.emc.com/leadership/digital-universe/2014iview/highvalue-data.htm
2. Cisco. Cisco Visual, Networking Index: Global Mobile Data Traffic Forecast 2016–2021 Q&A. https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-forecast-qa.html.
3. M. Shafiq, Z. Tian, A.K. Bashir, X. Du, M. Guizani, CorrAUC: a malicious bot-IoT traffic detection method in IoT network using machine learning techniques. Internet Things J. **PP**(99), 1 (2020). https://doi.org/10.1109/JIOT.2020.3002255
4. J. Qiu, Z. Tian, C. Du, Q. Zuo, S. Su, B. Fang, A survey on access control in the age of internet of things. IEEE Internet Things J. **7**(6), 4682–4696 (2020)
5. N. Hu, Z. Tian, H. Lu, X. Du, M. Guizani, A multiple-kernel clustering based intrusion detection scheme for 5g and iot networks. Int. J. Mach. Learn. Cybern. (2021). https://doi.org/10.1007/s13042-020-01253-w
6. Y.C. Hu, M. Patel, D. Sabella et al., Mobile edge computing—a key technology towards 5G. ETSI White Pap. **11**(11), 1–16 (2015)
7. W. Shi, J. Cao, Q. Zhang et al., Edge computing: vision and challenges. IEEE Internet Things J. **3**(5), 637–646 (2016)
8. Y. Sun, Z. Tian, M. Li, S. Su, X. Du, M. Guizani, Honeypot identification in softwarized industrial cyber-physical systems. IEEE Trans. Ind. Inform. **PP**(99), 1 (2021). https://doi.org/10.1109/TII.2020.3044576
9. C. Chen, B. Liu, S. Wan, P. Qiao, Q. Pei, An edge traffic flow detection scheme based on deep learning in an intelligent transportation system. IEEE Trans. Intell. Transp. Syst. **PP**(99), 1 (2020). https://doi.org/10.1109/TII.2020.3044576
10. S. Wan, X. Xu, T. Wang, Z. Gu, An intelligent video analysis method for abnormal event detection in intelligent transportation systems. IEEE Trans. Intell. Transp. Syst. **PP**(99), 1–9 (2020). https://doi.org/10.1109/TITS.2020.3017505
11. S. Ding, S. Qu, Y. Xi, S. Wan, Stimulus-driven and concept-driven analysis for image caption generation. Neurocomputing **398**, 520–530 (2020)
12. Y. Zhao, H. Li, S. Wan, A. Sekuboyina, X. Hu, G. Tetteh, M. Piraud, B. Menze, Knowledge-aided convolutional neural network for small organ segmentation. IEEE J. Biomed. Health Inform. **23**(4), 1363–1373 (2019)
13. S.D. Kamvar, M.T. Schlosser, H. Garcia-Molina, The eigentrust algorithm for reputation management in p2p networks. In *Proceedings of the 12th International Conference On World Wide Web (WWW), New York*, 2003, pp. 640–651
14. L. Xiong, L. Liu, PeerTrust: supporting reputationbased trust for Peer-to-Peer electronic communities. IEEE Trans. Knowl. Data Eng. **16**(7), 843–857 (2004)
15. M. Li, Y. Sun, H. Lu, S. Maharjan, Z. Tian, Deep reinforcement learning for partially observable data poisoning attack in crowdsensing systems. IEEE Internet Things J. **7**(7), 6266–6278 (2020)
16. C. Luo, Z. Tan, G. Min, J. Gan, W. Shi, Z. Tian, A novel web attack detection system for Internet of Things via ensemble classification. IEEE Trans. Ind. Inform. **PP**(99), 1 (2021). https://doi.org/10.1109/TII.2020.3038761
17. S. Moritz, E.N. Taoufik, E.W. Biersack, Long term study of peer behavior in the KAD DHT. IEEE/ACM Trans. Netw. **17**(5), 1371–1384 (2009)
18. Z. Tian, X. Gao, S. Su, J. Qiu, Vcash: a novel reputation framework for identifying denial of traffic service in internet of connected vehicles. IEEE Internet of Things J. **7**(5), 3901–3909 (2020)
19. J. Qiu, L. Du, D. Zhang, S. Su, Z. Tian, Nei-TTE: intelligent traffic time estimation based on fine-grained time derivation of road segments for smart city. IEEE Trans. Ind. Inform. **16**(4), 2659–2666 (2020)
20. Z. Tian, M. Li, M. Qiu, Y. Sun, S. Su, Block-DEF: a secure digital evidence framework using blockchain. Inf. Sci. **491**, 151–165 (2019)
21. N. Fedotova, M. Bertucci, L. Veltri, Reputation management techniques in DHT-based peer-to-peer networks. In *Second International Conference on Internet and Web Applications and Services (ICIW'07), Morne*, 2007, p. 4. https://doi.org/10.1109/ICIW.2007.53
22. H. Yu, M. Kaminsky, P.B. Gibbons, A.D. Flaxman, SybilGuard: defending against Sybil attacks via social networks. IEEE/ACM Trans. Netw. **16**(3), 576–589 (2008)
23. H. Yu, P.B. Gibbons, M. Kaminsky, F. Xiao, Sybillimit: a near-optimal social network defense against sybil attacks. IEEE/ACM Trans. Netw. **18**(3), 885–898 (2010)
24. C. Lesniewski-Laas. A sybil-proof one-hop DHT. In *Proceedings of the first Workshop on Social Network Systems (SocialNets'08), Glasgow*, 2008, pp. 19–24
25. N. Tran, B. Min, J. Li, L. Subramanian. Sybil-resilient online content voting. In *Proceedings of the 6th USENIX symposium on Networked systems design and implementation (NSDI'09), Berkeley*, 2009, pp. 15–28
26. Z. Gao, Y. Li, S. Wan, Exploring deep learning for view-based 3D model retrieval. ACM Trans. Multimed. Comput. Commun. Appl. **PP**(99), 1–18 (2020)
27. H. Lu, C. Jin, X. Helu et al., Research on intelligent detection of command level stack pollution for binary program analysis. Mobile Netw. Appl. (2020). https://doi.org/10.1007/s11036-019-01507-0
28. K. Walsh, E.G. Sirer. Experience with an object reputation system for peer-to-peer file sharing. In *Proceedings of the 3rd conference on Networked Systems Design & Implementation (NSDI'06). Berkeley*, 2006, pp. 1–14

Publisher's Note