Yang *et al. Journal of Cloud Computing: Advances, Systems and Applications* (2022) 11:2
https://doi.org/10.1186/s13677-021-00272-4

Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH Open Access

A trusted de-swinging k-anonymity scheme for location privacy protection

Manxiang Yang[1], Baopeng Ye[2], Yuling Chen[1*], Tao Li[1], Yixian Yang[3], Xiaobin Qian[4] and Xiaomei Yu[5]

Abstract

K-anonymity has been gaining widespread attention as one of the most widely used technologies to protect location privacy. Nevertheless, there are still some threats such as behavior deception and service swing, since utilizing distributed k-anonymity technology to construct an anonymous domain. More specifically, the coordinate of the honest node will be a leak if the malicious nodes submit wrong locations coordinate to take part in the domain construction process. Worse still, owing to service swing, the attacker increases the reputation illegally to deceive honest nodes again. To overcome those drawbacks, we propose a trusted de-swinging k-anonymity scheme for location privacy protection. Primarily, we introduce a de-swinging reputation evaluation method (DREM), which designs a penalty factor to curb swinging behavior. This method calculates the reputation from entity honesty degree, location information entropy, and service swing degree. Besides, based on our proposed DREM, a credible cloaking area is constructed to protect the location privacy of the requester. In the area, nodes can choose some nodes with a high reputation for completing the construction process of the anonymous domain. Finally, we design reputation contracts to calculate credit automatically based on smart contracts. The security analysis and simulation results indicate that our proposed scheme effectively resists malicious attacks, curbs the service swing, and encourages nodes to participate honestly in the construction of cloaking areas.

Keywords: k-Anonymity, De-swinging, Location privacy protection, Reputation evaluation, Smart contracts

Introduction

Location-Based Service (LBS) is a type of information service for mobile users based on information from mobile devices such as geographical location [1]. LBS has convenience and vulnerability since LBS provides services by node coordinates [2]. The attacker can easily deduce the user's identity [3, 4] and other sensitive personal information, even if the attacker is unaware of the user's status due to the compromise of location information. What's worse, the probability of disclosing the private information of honest nodes will increase if malicious nodes construct anonymous domains by dishonest means. The leakage of private location data in the LBS context has drawn significant attention from academics and industry on account of importance [5, 6]. To ensure the security of location privacy information, an increasing number of methods are being proposed by scholars [7–9]. The current method extensively used is the k-anonymity technology introduced from data privacy protection by Grutester et al. [10]. It can effectively protect private information by making the attacker cannot accurately distinguish the location information of request nodes from k locations.

Related works

Using k-anonymity, the attacker cannot correlate the query information of the requesting node by forming a cloaking area with the location information. Chow et al. [11] propose the first distributed k-anonymity scheme. To make k-anonymity more practically, Gang et al. [12] introduce location tags to distinguish the sensitive locations from ordinary locations for making the selected k locations scattered. For improving construction efficiency, Ge

*Correspondence: ylchen3@gzu.edu.cn
[1]State Key Laboratory of Public Big Data, College of Computer Science and Technology, Guizhou University, Guiyang, China
Full list of author information is available at the end of the article

et al. [13] present a location privacy protection scheme based on area awareness with the help of mobile service providers. Ghaffari et al. [14] propose a peer-to-peer solution of privacy protection query service. To resist attacks, Zhao et al. [15] present a k-anonymity scheme to prevent injection attacks on user locations. Ying et al. [16] design a distributed social-aware location privacy protection protocol to protect the original sender without the help of a trusted third party. Li et al. [17] propose a privacy-preserving LBS framework that supports the query area in a square area based on the user's location, and achieves fine-grained access control on the financial service provider data. Wang et al. [18] present a privacy metric and a utility metric to measure the performance of client-based personalized k-anonymity. Li et al. [19] and Wang et al. [20] design privacy protection methods to ensure the security of information exchange between users.

The distributed k-anonymity privacy protection schemes mentioned above all assume that users are honest, nevertheless, the users are self-interested in life. In response to this problem, Yang et al. [21] combine game theory and blockchain to construct a reputation mechanism based on the revenue function. Liu et al. [22] designed a scheme with the help of blockchain to restrain users' self-interest by punishing their dishonest behaviors. Fortunately, Li et al. [23, 24] and Wang et al. [25] tackle the problem of malicious attacks on the blockchain. lu et al. [26] develop a credibility assessment algorithm based on both direct historical transactions and indirect observations of nodes to combat the spread of fraudulent information. Li et al. [27] uses a probability threshold to reflect the user's reputation, and only initiate the construction of a cloaking area when the reputation of the requesting node reaches the threshold. Luo et al. [28] and Li et al. [29] devise the trust management method such that both the requester and the cooperator will only cooperate with the nodes they trust. Li et al [30] introduce a reputation management algorithm to optimize the k-anonymity technique. To encourage users to assist, Yang et al. [31] adopt a single-round sealed double auction mechanism, that allows multiple request nodes to obtain the actual location of the cooperators. Li et al. [32] pointed out that the above schemes need a credible auctioneer then presented a distributed k-anonymity privacy protection scheme based on reputation incentives. In addition, Chow et al. [33] and Gong et al. [34] indicated that the pseudonym is replaced for better privacy. To enhance the credibility of the privacy protection scheme, Yuan et al. [35] propose a privacy-preserving framework without online trusted third parties that can protect the locations of workers and tasks while keeping the distance-aware information on the protected locations.

Different from our proposal, the above works only decrease the probability of user dishonesty by reducing the user's reputation. They fail to recognize that malicious users can quickly achieve a high reputation through a series of honest activities to deceive honest users again. The problem reduces the cost of fraud and allows the location information of honest users to be exposed multiple times. An attacker analyzes the correlation of multiple locations to deduce the user's intimate location information, such as the home address, workplace, etc. In addition, our proposal is more adapted to practical applications as the entropy of the user's current location information is calculated.

Motivations and contributions

The existing schemes about distributed k-anonymous cannot protect the location privacy of nodes completely. Namely, these schemes still have matters with behavior deception and service swings. Concretely, it is difficult to judge malicious requests, which initiate normally in the location of low request probability at midnight, without considering location information entropy. What's more, malicious nodes have more opportunities to attack honest nodes if there is service swinging in reputation-based schemes. As shown in Fig. 1, the left-hand side shows the locations where users frequent from Monday to Friday. In the morning, the user travels from A to B and then to C. In the afternoon, the user returns to A after passing through B from C. The regularity of the user's appearances makes it easy for an attacker to infer the sensitive location based on multiple observations. Most likely, A is the user's home, B is his child's school and C is the company. The attacker could extort the user or stole from the house while the user is away on business or in hospital. These attacks are a threat to the user's personal and property security. In this paper, we propose, to ensure the privacy information of honest nodes, a trusted de-swinging k-anonymity scheme for location privacy protection. Firstly, we design a DREM, in that a penalty factor is set to curb the swinging behavior. Meanwhile, we build trustworthy anonymous domains based on smart contracts with the DREM. In the area, both the requesting node and the cooperative node only cooperate with their reliance. Then, reputation contracts in view of smart contracts are designed, which avoid information leakage caused by third-party attacks, to calculate nodes' reputation automatically. Finally, security analysis and simulation results show that our scheme can effectively curb the swinging behavior of malicious nodes, resist attacks and encourage nodes to participate honestly in the construction of cloaking areas. In a nutshell, the main contributions are as follows.

- We design a DREM, which encourages nodes to participate sincerely in the construction of cloaking areas from entity honest degree, location information entropy, and service swinging. Besides, we design a

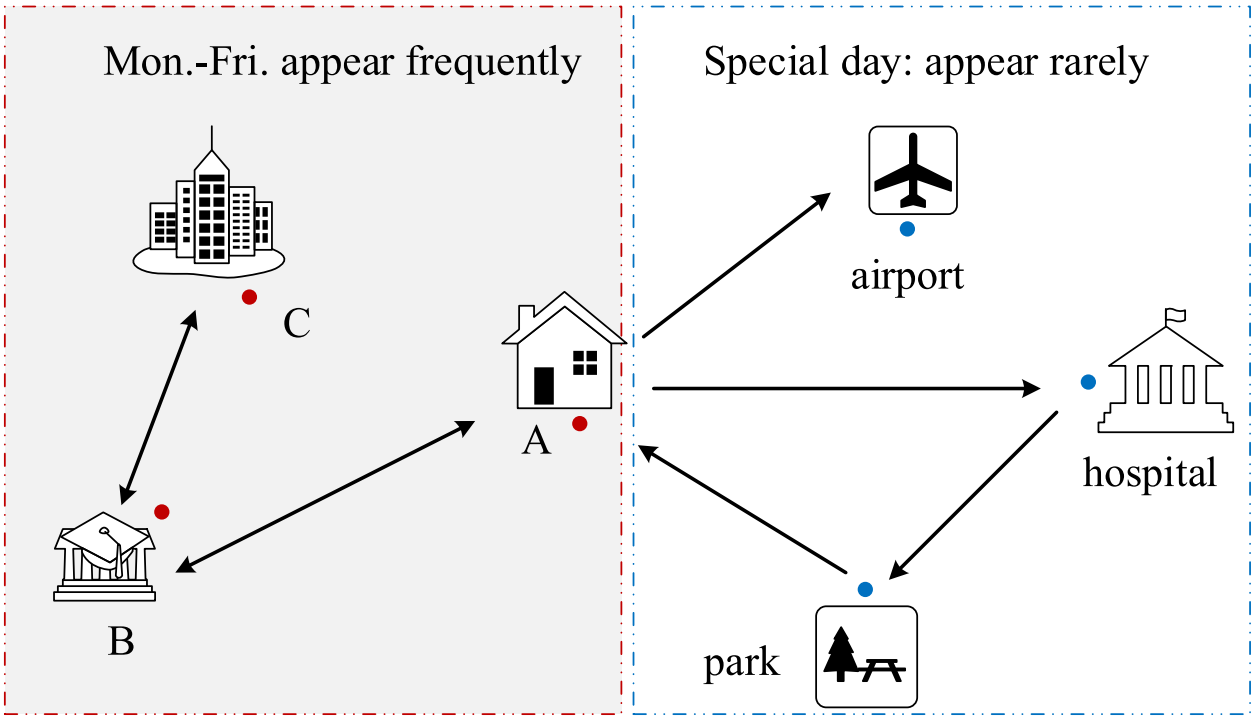

Fig. 1 The location privacy leakage caused by the malicious users

punishment factor according to the swinging degree to curbs the malicious behaviors.
- We structure a trusted cloaking area dependent on the DREM, that protects the location privacy of request nodes and cooperators. In the process of construction, both request nodes and assist nodes only cooperate with their reliance that selecting by the smart contracts.
- We design reputation contracts based on smart contracts, which automatically execute reputation calculation algorithms and store the scores in the blockchain, to make the processes of calculation decentralized, open, and transparent.
- The security analysis and simulation results indicate that our proposed scheme effectively curbs the swinging behavior of malicious nodes, resists attacks, detects the malicious nodes quickly, and encourages nodes to participate honestly in the construction of cloaking areas.

Roadmap

The organization of this article is as follows. We start with pinpointing some preliminaries like k-anonymity technology and de-swinging in the "Preliminaries" section. The scheme is presented in the "A trusted de-swinging k-anonymity scheme for location privacy protection" section. The security and simulation analysis are introduced in the "Security & simulation analysis" section. Finally, we conclude this article in the "Conclusion" section and prospect the future works.

Preliminaries

Distributed k-anonymity

K-anonymity is a popular location privacy protection technology that associates the requester's location with at least $k-1$ locations of assistants to construct an anonymous domain. In the area, attacker cannot associate a query with a specific participant with a probability greater than $1/k$. The technology was applied initially to data fusion [36], data analysis, and data prediction [37]. It was introduced into location privacy protection, such as IoT [38, 39], vehicle networks [40, 41] as its technology evolve. In general, existing privacy-preserving schemes for k-anonymity fall into two main categories: centralized, which requires the help of a third-party server, and distributed, which achieves anonymity through the assistance of participants. The centralized k-anonymity suffers from problems such as single points of failure and privacy disclosure by third-party servers. Conversely, distributed k-anonymity has become a popular field of research in recent years because it overcomes the disadvantages of centralized schemes. Our scheme is designed based on

distributed k-anonymity. The process of construction are as follows.

1) The requesting node broadcasts a request for cloaking area construction.
2) Assistance nodes respond to the requester and send their locations to the requesting node.
3) The requesting node constructs the cloaking area with the help of $k-1$ locations of assistance nodes.
4) The requesting node sends the cloaking area to the location service provider (LSP).
5) The LSP seeks the results according to the cloaking area and query information submit by requesting node.
6) The requesting node selects the query results according to their real location after receiving the returned results.

De-swinging

There are dynamic swing attacks in the reputation-based k-anonymity privacy protection scheme, that is, the behavior of malicious nodes change between honestly and maliciously. In other words, malicious nodes attack after accumulating a high trust value by interacting honestly. After that, they stop transacting until the impact of the last malicious behavior becomes weak due to the time attenuation factor. At this time, malicious nodes conduct multiple successful transactions within a short period to increase the reputation quickly then attack again. Malicious users rapidly increase their reputation after it has been reduced in this way. Worse still, malicious users repeatedly trick honest users with high reputations, so that honesty users have a high probability to assist malicious users in building anonymous domains. The location privacy of the honest user is compromised repeatedly after being maliciously attacked. To make k-anonymity robust against internal attacks [42] and protect location track information [43] better, we propose a de-swinging method, which designs a penalty factor according to the swing degree of the malicious nodes. The realization process of de-swinging is as follows.

1) Calculating the difference of node's latest two reputations $\Delta c' = |LC_{i+1} - LC_i|$.
2) If $\Delta c' > C$, the node is malicious one, to calculate $\Delta c'' = \Delta c'/LC_{i+1}|$.
3) Setting penalty factor according to $\Delta c''$.
4) Calculating reputation with the penalty factor.

A trusted de-swinging k-anonymity scheme for location privacy protection

In this section, we present the framework of our scheme, the reputation evaluation method, the process of constructing anonymous domains, and the reputation contract algorithm.

System architecture

To better construct the trusted clocking area, the system framework of our scheme is depicted in Fig. 2, which

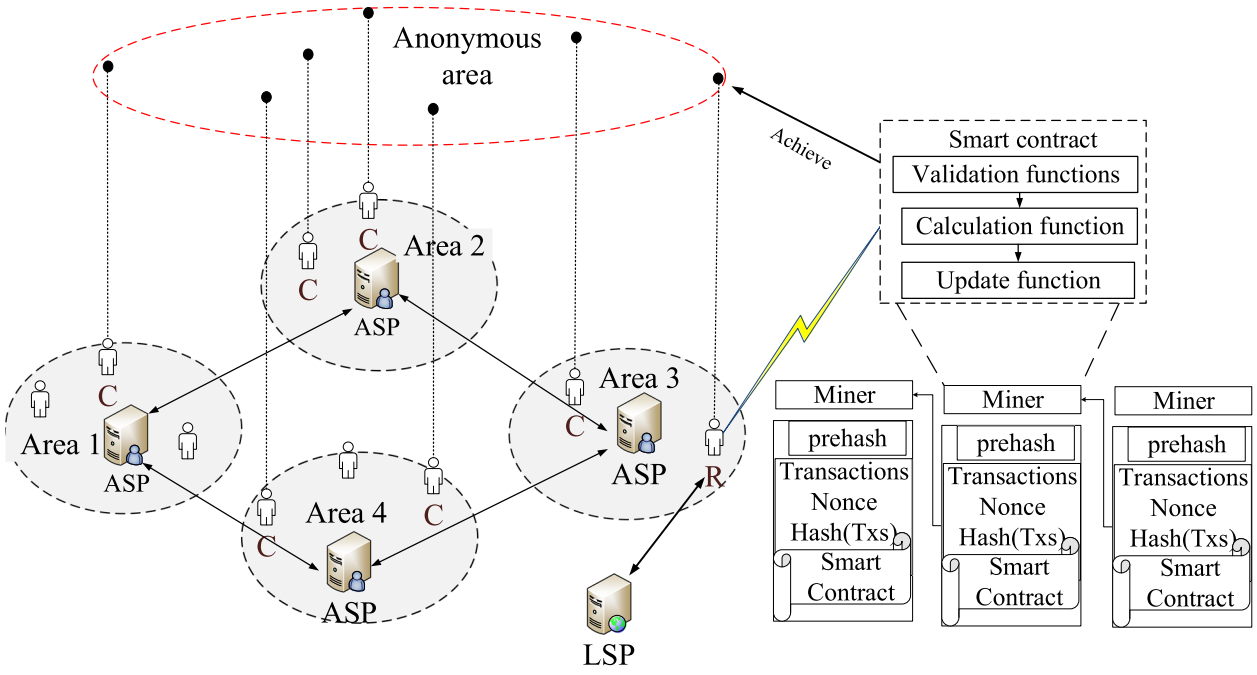

Fig. 2 System architecture

consists of four parts, anonymous service provider (ASP), LSP, reputation contracts, and nodes. R represents request nodes, and C stands for assist nodes.

ASP is responsible for allocating pseudonym *id* and asymmetric key pairs (P_k, S_k) to provide preparation and security for nodes during the process of construction. ASP randomly generates a pseudonym *id* for the nodes When they register. The Num_{id}, which is increased by one after the user initiates or participates in the construction of a cloaking area, records the number of times the id is used. The ASP also assigns a pair of asymmetric public-private key pairs (P_k, S_k) to the nodes after they registered successfully, which are only saved by the nodes. The nodes sign their information with the S_k before they send information to the reputation contracts, then the smart contracts verify whether the information has tampered with the P_k. Table 1 shows the data definition stored in ASP.

LSP provides location-based services for nodes, such as querying nearby restaurants, hotels, etc. The process for querying the service is as follows. Firstly, the requesting node broadcast the request for cloaking area construction. Then, the requesting node constructs the cloaking area $P_{se}\left\{Locr_0, Locr_1, ..., Locr_{k-1}\right\}$ after receiving the location of $k-1$ assistance nodes. Lastly, the LSP query the results according to the Pse and query information submit by requesting node. The requesting node selects the query results according to their real location $Locr_0$ after receiving the returned results.

Reputation contracts are designed based on smart contracts to calculate reputation. We design three reputation contract functions: verification function, reputation calculation function, and update function. The verification function is applied to verify the identity of nodes. The calculation function selects the assistants whose reputation is equal or greater than the request threshold after calculates the reputation according to the DREM. The update function is responsible for updating the scores timely.

The nodes can be a cooperator or a requester. While a node needs location services, it can be a requesting node to initiate a cloaking area construction request for obtaining location services while protecting location privacy. After receiving the cloaking area construction request, the nodes can cooperate with the construction as a cooperator.

Table 1 The data definition stored in ASP

Notation	Definition
User	The real name of nodes
id	The current pseudonym
Num_{id}	The number of pseudonym is used
id_list	The list of historical pseudonyms

De-swinging reputation evaluation

The de-swinging reputation evaluation method calculates the reputation from three aspects: entity honesty degree, location information entropy, and service swinging degree. We adopt the model combining entities and data to calculate the reputation of nodes. On the one hand, we calculate the entity honesty degree with the historical scores for encouraging nodes to participate genuinely in the construction of cloaking areas. On the other hand, we evaluate the reliability of current data from the probability of prior location, the location request probability, and the time request probability. We call the reliability of current data like location information entropy. In addition, the malicious nodes deceive other nodes after accumulating a high reputation. What's worse, they will attack again while the reputation rebounds to a high level. For this reason, a penalty factor is designed according to the swing degree to curb users' service swing.

We divide the use's reputation into δ levels $\{Lev_1, Lev_2, ..., Lev_\delta\}$, Lev_1 indicates the lowest reputation level, Lev_δ indicates the highest reputation level. The data definition storage in blockchain is shown in Table 2.

Entity honesty

We adopt the latest historical request scores $\{LR_1, LR_2, ..., LR_n\}$ and historical assistance scores $\{LC_1, LC_2, ..., LC_n\}$ to calculate entity honesty. We introduce the forgetting function show as Formula (1) to calculate the honesty of nodes. G_R, G_C represents the forgetting value that calculate from the current time T_x to the corresponding update time $TimeLR_n$ and $TimeLC_n$ of the reputation LR_n and LC_n. q represents the time attenuation factor. W is the forgetting cycle. As shown in Formula (3) and Formula (4), the closer the scoring time to the current calculation time, the greater the impact on the honesty of the entity. E^R_{hon}, E^C_{hon} indicate respectively the honesty of the entity integrity degree request nodes and assistance nodes. LR^i_R represents the requesting score of the request node when the node as a request node. LC^j_R is the assistance score of the request node while the node as a cooperator. LR^i_C is the request score of assistance node when the node as a

Table 2 The data definition stored in the blockchain

Notation	Definition
id	The node pseudonym
LR	The list of historical request scores
LC	The list of historical assistance scores
LR_Time	The update time of request score
LC_Time	The update time of assistance score
n^R_{pro}	The number of requests
n^C_{pro}	The number of assistance

request node. LC_C^j is the assistance score of the assistance node when the node as a cooperator.

$$G_R = exp\left\{-q\left(T_x - TimeLR_n/W\right)\right\} \tag{1}$$

$$G_C = exp\left\{-q\left(T_x - TimeLC_n/W\right)\right\} \tag{2}$$

$$E_{hon}^R = \alpha \sum_{i=1}^{n} LR_R^i \times G_R + (1-\alpha) \sum_{j=1}^{n} LC_R^j \times G_R \tag{3}$$

$$E_{hon}^C = \alpha \sum_{i=1}^{n} LR_C^i \times G_C + (1-\alpha) \sum_{j=1}^{n} LC_C^j \times G_C \tag{4}$$

Location information entropy

Location information entropy refers to the uncertainty of location information. The higher the location information entropy, the higher the reliability of the current location.

If the nodes structure cloaking area in mountains, rivers, or deserts, the probability of being honest is low. We divide the current area into y categories according to location characteristics $\left\{typ_{loc}^1, typ_{loc}^2, ..., typ_{loc}^y\right\}$, the credible probability of each type of location is $P_{y-Ty}\left\{P_{y-Ty}^1, P_{y-Ty}^2, ..., P_{y-Ty}^y\right\}$.

We believe that the location where queries have occurred is more reliable. As shown in Fig. 3, we divide the entire location space into many grids. White indicates that the request probability is zero, black means the request probability is high, which is 0.87 in the figure. According to the historical query data in each grid, the location request probability of every grid is $P_{l-loc}\left\{P_{l-loc}^1, P_{l-loc}^2, ..., P_{l-loc}^l\right\}$.

The time when the nodes initiate a request has a regularity, the requesting which launched during a period with a high request probability is more reliable. By counting the number of requests, we can learn the historical habit of initiating the request. We divide a day into t periods $\left\{tim_{loc}^1, tim_{loc}^2, ..., tim_{loc}^t, \right\}$, the request probability for each period is $p_{t-tim}\left\{p_{t-tim}^1, p_{t-tim}^2, ..., p_{t-tim}^t\right\}$.

When calculating the location information entropy of the cooperator, only its true probability of prior location is considered, as shown in Formula (5). The location information entropy S_{loc}^R of the request node is calculated from three aspects, as shown in Formula (6).

$$S_{loc}^C = -\varphi \sum_{j=1}^{y} p_{y-Ty}^j \log_2 p_{y-Ty}^j \tag{5}$$

$$\begin{aligned} S_{loc}^R = {} & (-\tau) \sum_{j=1}^{y} p_{y-Ty}^j \log_2 p_{y-Ty}^j \\ & + (-\phi) \sum_{j=1}^{l} p_{l-loc}^j \log_2 p_{l-loc}^j \\ & + (-\mu) \sum_{j=1}^{t} p_{t-tim}^j \log_2 p_{t-tim}^j \end{aligned} \tag{6}$$

Service swing

To curb the swing behavior of malicious nodes, we store the swing behaviors with one-dimensional array $b[n]$ and $a[m]$, which is a basis to calculate penalty factors. Taking the calculation process of assistance nodes' swing degree a[m] as an example. Firstly, move the elements of $a[0]$ one bit to the right, making it empty. Then calculate the difference between the last two historical assistance scores as the formula $\Delta c'$. Finally, calculate the ratio of $\Delta c'$ to the latest historical score with the formula $\Delta c''$. To set $a[0]$ according to $\Delta c''$, as shown in Table 3.

The node is judged as a swing one when $\Delta c''$ is greater than 0.4. To set $a[0]$ differently according to $\Delta c''$, the

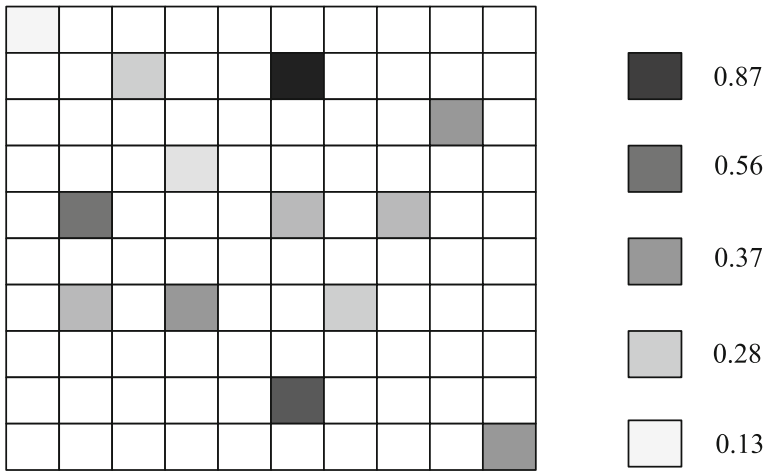

Fig. 3 The distribution of historical query probability

Table 3 The mapping from $\Delta c''$ to $a[0]$

$\Delta c''$	$a[0]$
$\Delta c'' \leq 0.25$	$\Delta c''$
$0.25 < \Delta c'' \leq 0.3$	0.6
$0.3 < \Delta c'' \leq 0.4$	1
$0.4 < \Delta c'' \leq 0.6$	2

greater the degree of swing, the greater $a[0]$. For curbing the swing behavior, we designed the penalty factor Q^R_{pun}, Q^C_{pun}, for request nodes and cooperators. The calculation method is as Formula (7) and Formula (8), the smaller the punishment factor, the greater the punishment. λ and δ are the parameters that record the malicious behavior of users as assistance and requests.

$$Q^C_{pun} = exp\left\{\sum_{k=0}^{n} \lambda\left(d^{-k} \times a[k]\right)\right\} \tag{7}$$

$$Q^R_{pun} = exp\left\{\sum_{k=0}^{n} \delta\left(d^{-k} \times b[k]\right)\right\} \ (d, \lambda, \delta \geq 1) \tag{8}$$

It can be seen from the above formula that more weight is given to the recent swing behavior. The $\Delta c'$ becomes larger and the penalty factor decreases when the recent scores fluctuate greatly. when $a[k] < 1$, $\lambda = 1$, while $a[k] \geq 1$, λ increases with the increase of $a[k]$, leads the penalty factor to increase slowly after the malicious transaction.

$$F^R_{cre} = exp\left(Q^R_{pun}\right) \times \ln\left(\beta \times E^R_{hon} + \eta \times S^R_{loc}\right) \tag{9}$$

$$F^C_{cre} = exp\left(Q^C_{pun}\right) \times \ln\left(\beta \times E^C_{hon} + \eta \times S^C_{loc}\right) \tag{10}$$

As shown in the Formula (9) and Formula (10), the reputation of nodes F^R_{cre}, F^C_{cre} is proportional to the entity honesty E^R_{hon}, E^C_{hon} and the location information entropy S^R_{loc}, S^C_{loc}. The reputation also refers to the penalty factor Q^R_{pun} and Q^C_{pun}. β and η are weighting parameters. When nodes attack with high reputations, reputation value will drop rapidly because of the decrease of the penalty factor. If the malicious node builds a cloaking area honestly after an attack, the reputation will increase slowly. Furthermore, if the score rises too fast, the penalty factor will decrease, and the reputation will decrease accordingly. Therefore, our scheme can curb the swing behavior of malicious nodes, encourage nodes to participate honestly in the construction process of cloaking areas.

Trusted cloaking area

We construct an anonymous domain based on the DREM and smart contracts. The process of cloaking area construction is shown as Fig. 4.

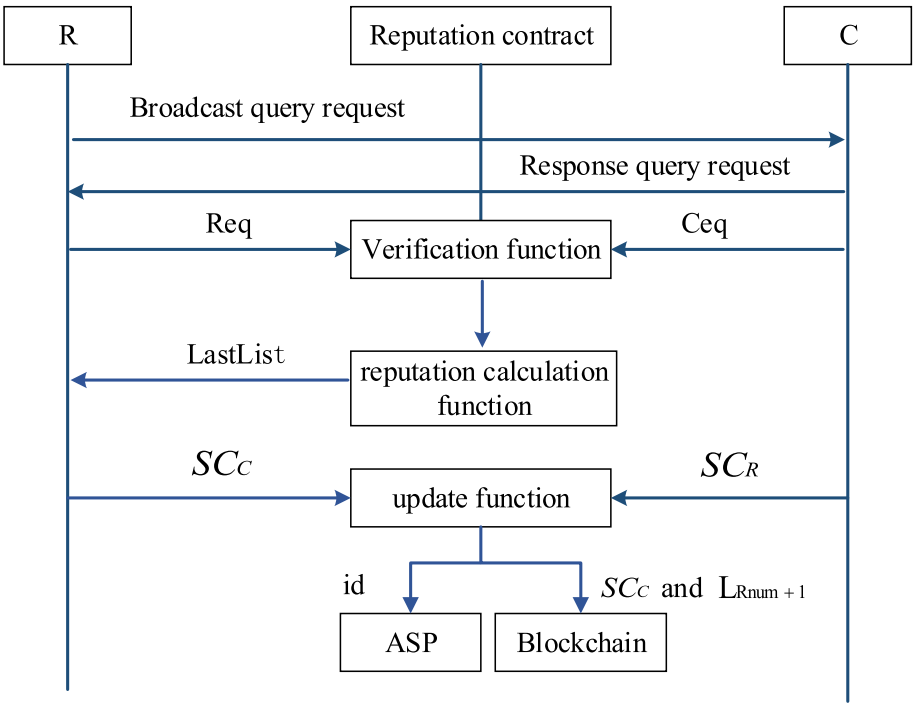

Fig. 4 The interaction process of anonymous domain construction

At the beginning of the process, the request node sends the request information $Req\{R_{id}, Num_{LR}, Pk_R, M_R, k, Sig_{PkR}(Num_{LR}||M_R)\}$ to the smart contracts after broadcast the construction request. For responding the request, the collaborator sends the assistance information $Req\{C_{id}, Num_{LC}, Pk_C, M_C, Sig_{PkC}(Num_{LC}||M_C)\}$ to the smart contracts. R_{id} and C_{id} are the pseudonyms of nodes; Num_{LR}, Num_{LC} represent the transaction index number, that record the latest historical information in the blockchain. The Pk_R, Pk_C as the public key of nodes. M_R, M_C as the reputation threshold. k is the number of assistance nodes. $Sig_{PkR}(Num_{LR}||M_R)$, $Sig_{PkC}(Num_{LC}||M_C)$ as the signature to the information.

We calculate the degree of assistance $H_{pro} = n^C_{pro}/\left(n^C_{pro} \dot{+} n^R_{pro}\right)$ of the request node after the smart contracts verifies the signatures with the public keys Pk_R and Pk_C. n^R_{pro} is the number of requestion, n^C_{pro} is the number of cooperation. If H_{pro} is higher than the system value, the reputation value F^R_{cre} of the request node is calculated according to the DREM. If $F^R_{cre} \geq M_C$, then the corresponding assistance node C_{id} is recorded in the candidate list *PreList*. Then calculating the reputation of assistance nodes in *PreList*, if $F^C_{cre} \geq M_R$, then record the C_{id} in the cloaking area construction list *LastList*.

The smart contracts send the location information $\{Locr_1, Locr_2, ..., Locr_{k-1}\}$ of the collaborator in *LastList* to the request node. With the help of the $k-1$ locations, the request node constructs the cloaking area $P_{se}\{Locr_0, Locr_1, Locr_2, ..., Locr_{k-1}\}$ then sends the query information to the *LSP*, which returns the query content to the requesting user.

The request node to score $k-1$ cooperators separately and signatures with Sk_R after the cloaking area is constructed successfully. The cooperators to score the request node and signature with Sk_C. Then, they send the scores of cooperators $SC_C\left\{\left(SC^1_C, Sig_{SKR}\left(SC^1_C\right)\right), ..., \left(SC^{k-1}_C, Sig_{SKR}\left(SC^{k-1}_C\right)\right)\right\}$, and the scores of requester $SC_R\left\{\left(SC^1_R, Sig_{SKC}\left(SC^1_R\right)\right), ..., \left(SC^{k-1}_R, Sig_{SKC}\left(SC^{k-1}_R\right)\right)\right\}$ to the smart contracts, which calculate the reputation score $LR_{num} + 1 = \sum_1^{k-1} SC^i_R/(k-1)$ of the request node.

Reputation contracts

Based on the smart contracts, we design reputation contracts that include verification function, reputation calculation function, and update function.

Verification function: we verify the identity information of the node who participates in the construction process before calculating the reputation to ensure the legality of nodes. Other functions can only be invoked if the identity of both the assisting user and the requesting user has been successfully verified. The verification function is shown in algorithm 1.

Algorithm 1: Verification function

Data: the information of requester*Req*,the information of k-1 collaborators *Ceq*

```
R ← Req;
C_1, ..., C_{k-1} ← Ceq_1, ..., Ceq_{k-1};
if (PK_R(Sig_PkR))==(Num(LR)||MR) then
    if (PK_C(Sig_PkC))==(Num(LC)||MC) then
        go to the calculate credit function;
    else
        Reject the request;
    end
else
    Reject the request;
end
```

The calculation function: as algorithm 2 shows, we calculate the reputation of requester and collaborators with the help of DREM by entity honesty, location information entropy, and penalty factor. This function takes as input the historical scores of the requester and k-1 assisters. The output of this function is a list of assisted users.

The update function: the contract calculates the final score LRnum+1 of the requester after receiving the scores of k-1 assistants and requesters. Subsequently, the nodes with the highest reputation update the score list. The update function is as shown in Algorithm 3.

Security & simulation analysis

In this section, we will validate the safety and effectiveness of the proposed scheme. We organize this section by the analysis of the safety analysis and simulation results.

Security analysis

Anti-attack analysis

We analyze the node attack from the following aspects: 1) malicious nodes initiate or participate in the construction of cloaking areas with fake geographic locations; 2) malicious nodes adopt a swing strategy to increase the reputation in a short period after an attack. To prevent the first type of attack, we calculate the location information entropy, which is adopted to evaluate the authenticity of the location, to ensure that nodes participate in anonymous domain construction with their true locations. For the second type of attack, we design the penalty factor Q^R_{pun}, Q^C_{pun} based on the swing degree. Once the nodes act maliciously, their reputations will rapidly decrease. Even if the malicious nodes accumulate a high score in a short period, the penalty factor will

Algorithm 2: Calculate credit function

Data: requestor's history scores LR_r and LC_r,historical scores for $k-1$ helpers $LR_{c1}, ..., LR_{ck}$and $LC_{c1}, ..., LC_{ck}$,
Locationinformation $p_{y-Ty}, p_{l-loc}, p_{t-tim}, n^C_{pro}, n^R_{pro}$
Result: list of assist users Lastlist($Cid_1, ..., Cid_{k-1}$)

$H_{pro} \leftarrow n^C_{pro}/n^R_{pro} + n^C_{pro}$;
if H_{pro}*greater than system threshold* **then**
  $E^R_{hon} \leftarrow x\left(LR^i_R+, ..., +LR^n_R\right) + (1-x)\left(LR^j_C+, ..., +LR^n_C\right) exp(-q * \Delta T\ /W)$;
  $E^{C1}_{hon}..., E^{Ck-1}_{hon} \leftarrow \left\{x\left(LR^i_C+, ..., +LR^n_C\right) + (1-x)\left(LC^j_C+, ..., +LC^n_C\right)\right\} exp(-q * \Delta T\ /W)$;
  $S^C_{loc} \leftarrow (-O)\left(p^1_{y-Ty} log2p^1_{y-Ty}+, ..., +p^y_{y-Ty} log2p^y_{y-Ty}\right)$;
  $S^R_{loc} \leftarrow (-O)\left\{\left(p^1_{y-Ty} log2p^1_{y-Ty}+, ..., +p^y_{y-Ty} log2p^y_{y-Ty}\right) + \left(p^1_{l-loc} log2p^1_{l-loc}+, ..., +p^l_{l-loc} log2p^l_{l-loc}\right) + \left(p^1_{t-tim} log2p^1_{t-tim}+, ..., +p^t_{t-tim} log2p^t_{t-tim}\right)\right\}$;
  $Q^R_{pun} \leftarrow exp\left(g\left(d^{-1} * b[1] + d^{-2} * b[2] + ..., + d^{-m} * b[m]\right)\right)$;
  $Q^C_{pun} \leftarrow exp\left(h\left(d^{-1} * a[1] + d^{-2} * a[2] + ..., + d^{-n} * a[n]\right)\right)$;
  $F^R_{cre} \leftarrow exp\left(Q^R_{pun}\right) ln\left(E^R_{hon} * g + j * S^R_{loc}\right)$;
  $F^C_{cre} \leftarrow exp\left(Q^C_{pun}\right) ln\left(E^C_{hon} * g + j * S^C_{loc}\right)$;
  if F^R_{cre} *greater than* $C_i.M_C$ *and* F^C_{cre} *greater than* $R.M_R$ **then**
    $Lastlist \leftarrow Cid_i$
  end
else
  Insufficient assistance, can'trequest
end

Algorithm 3: Update function

Data: The score of requestor $SC_R = \left\{SC^1_R, ..., SC^{k-1}_R\right\}$, The score of collaborator $SC_C = \left\{SC^1_C, ..., SC^{k-1}_C\right\}$
Result: L_{Rnum+1}, SC_C

if *The requestor accept* SC_R **then**
  $L_{Rnum+1} = SC^1_R + ..., +SC^{k-1}_R/k-1$;
  send the Rid to ASP;
else
  To score again
end
if *The collaborator accept* SC_C **then**
  Broadcast to other users;
  send the Cid to ASP;
else
  To score again
end

inhibit the increase of the reputation to avoid the swing behavior.

Hitchhiking attack

Free-riding behavior means that the node only enjoys assistance services provided by other users, refuses to afford cooperation for other users. To solve this problem, we set the degree of assistance H_{pro} to measure the assistance probability of the requester. Only the H_{pro} is greater than the threshold, can the cloaking area construction be initiated.

Optimal times of pseudonyms usege

If the attacker obtains a node's location with the same pseudonym multiple times, he will infer the identity of this node based on the associated information. Therefore, the pseudonym needs to be replaced after a certain number of uses. We take the cooperators as an example to infer the optimal use times g of pseudonyms. Assuming the probability that the assistance node in the area where the request node intends to initiate a request and assists the request node is p_s, the attacker has obtained the location privacy information of the cooperator h times, the probability of inferring the identity information of the cooperator is p_a. During the period of the same pseudonym, the attacker can infer the possibility of the cooperator's private information is: $P_{t,c} = C^h_g p^h_s p_a$. To ensure the security of the privacy information, $P_{t,c} < W$, W tends to 0. If $h = 4$, $p_s = 0.16$, $p_a = 0.65$, $W = 0.1$, then the optimal times of usage is $g = 10$.

We compare our scheme with existing works and the results are shown in Table 4 where we can easily identify the difference in the proposed scheme.

Table 4 Performance comparison

Property	Literature 28	Literature 31	Literature 27	Our scheme
De-swinging	×	×	×	✓
Hitchhiking attack	×	×	✓	✓
Location Information Entropy	×	×	×	✓
Optimal times of pseudonyms usege	✓	×	×	✓

Table 5 The parameter settings

Parameter	Value	Definition
n	10	The number of historical reputation participating in calculation
m	15	The length of swing array
q	0.5	The time decay factor
w	15	The time decay period
α	0.46	The entity honesty parameter
λ	3.8	The malicious behavior recording parameters of cooperators
δ	1.6	The malicious behavior recording parameters of request node
β	0.63	The weight of entity honesty

Simulation analysis

We divide the nodes' reputation into 10 levels $\{Lev_1, Lev_2, ..., Lev_{10}\}$, the minimum reputation level Lev_1 is 1, the highest reputation level Lev_{10} is 10. The definite parameters are set in Table 5.

Reputation value

We compare our scheme with Lu et al. [26] and Li et al. [30] in terms of trends in reputation. As shown in Fig. 5, the black line represents the scheme of Li, the blue line shows the scheme of Lu, our scheme is indicated by the red line. Before the 11th round, the reputation increased rapidly because they initiated and participated honestly in the construction of the cloaking area. However, the reputation drops after the 12th round due to malicious behavior. In their scheme, the reputation drops to the lowest only after the nodes commit evil multiple times because they lack consideration of the effect of service swing on reputation, which will give the nodes multiple opportunities to be malicious. Furthermore, the growth rate of reputation after nodes behave maliciously is the same as before they are evil, and nodes can rapidly increase the reputation in a short period to commit evil again in succession. On the contrary, owing to malicious behavior, the reputation of our scheme drops extremely in 12 rounds because of the malicious behavior, which keeps the future growth rate of reputation lowly for a period. Even if the malicious nodes cheat a high score, the growth slope of reputation is much lower than the previous 11 rounds, curbing the swing behavior. Figure 6 shows that the reputation of honest nodes increases with the escalation of the number of rounds. There is a problem in Lu et al. [26] and Li et al. [30] that nodes can reach the maximum reputation in a short period, which will increase the probability of malicious behavior. To prevent nodes from exploiting high reputation values for evil purposes, the nodes' reputation will grow at a slow rate after reaching a threshold in our scheme. Besides, our reputation mechanism can capture very subtle changes. For example, in the 15th and 19th rounds, the nodes are in a position where the probability of a request is lower, their reputation drops accordingly.

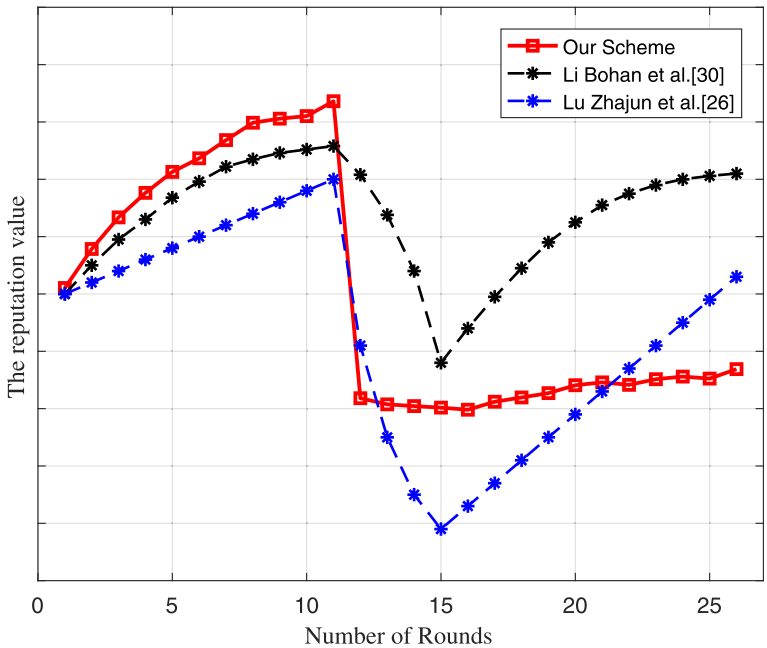

Fig. 5 The reputation value tendecy of malicious nodes compare with [26] and [30]

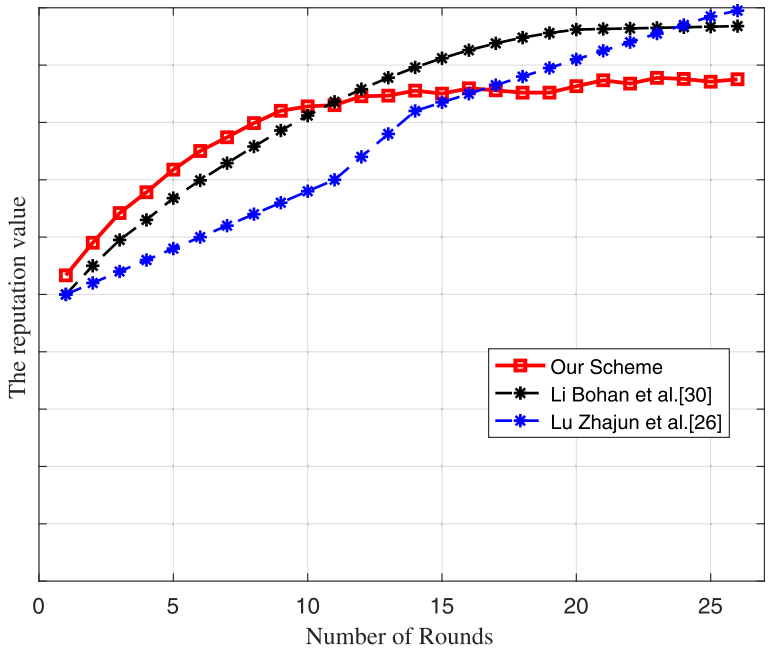

Fig. 6 The reputation value tendecy of honest nodes compare with [26] and [30]

Figures 7 and 8 show the reputation of nodes who play two roles: request node and cooperator. Figure 7 shows the reputation of the node, who requests maliciously while assisting honestly. The reputation of the request decreased rapidly in the 11th round due to malicious requestion. Though the node is honest when assisting, the reputation of assistance is decreased in the 11th round and then increased slowly during subsequent honest assistance. In this way, our scheme can reduce the probability that the node initiates a request successfully, stabilize the number of cooperators, and ensure the speed and quality of the clocking area construction. Figure 8 shows the reputation of the node, who assist maliciously while request honestly. The reputation of assistance and request

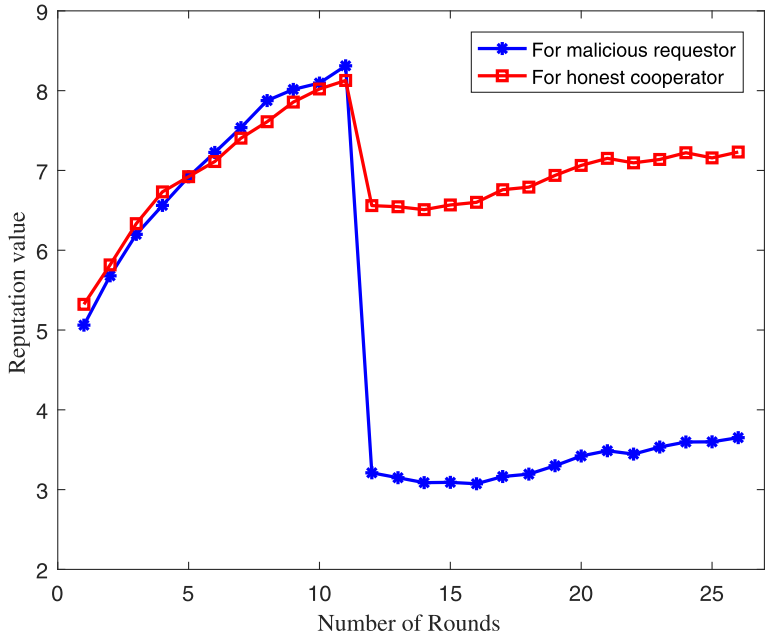

Fig. 7 The reputation of malicious request nodes

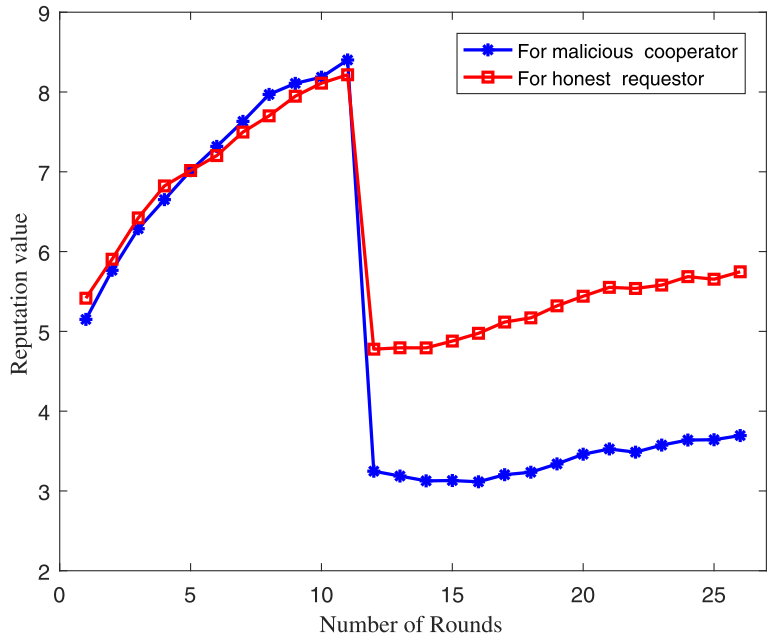

Fig. 8 The reputation of malicious cooperate nodes

both decrease rapidly because of malicious assistance and increase slowly hereafter. Therefore, our scheme restricts the assisting behavior also reduces the probability of the request successfully if the node attacks while assisting.

Efficiency

We set a specific scenario to verify the effectiveness of detecting malicious nodes. Usually, nodes are more likely to attack after the reputation reaches a certain point: 1) be honest in the earlier stage, and be malicious once the reputation rises to a certain level; 2) apply dynamic swing behaviors between honesty and maliciousness. In the first case, the nodes are honest in the previous η round for improving the reputation quickly. They are malicious after η round, $\eta' < \eta < \eta''$. Hypothesis $\eta' = 5$, η''=20, the detection results are shown in Fig. 9. The nodes increase their reputation rapidly in the first 5 rounds, then start to

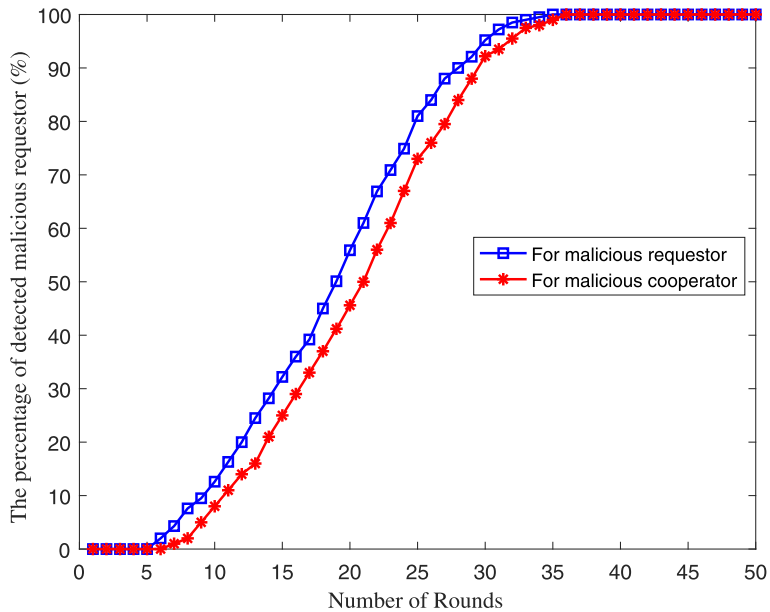

Fig. 9 The percent of case 1 detected malicious nodes

attack in the 6th round. In the 30th round, the percentages of malicious request nodes and malicious cooperators detected respectively were 97.13% and 93.5%.

The percentage of malicious nodes detected in the second category is shown in Fig. 10. Assume that the alternating cycle of the nodes is malicious and honest is θ, $5 < \theta < 15$. In the 30th round, the percentage of malicious request nodes and malicious assistance nodes detected is 81.13% and 70.33% respectively. The node can be marked as a malicious one when its swing is greater than the value set by the scheme. Obviously, it takes longer to detect all malicious nodes than the first type because they adopt the swing strategy. It can be seen that our scheme can detect malicious nodes with high accuracy. Our scheme can detect malicious nodes with a high probability after the 30th round, and build an anonymous domain stably with trusted nodes.

To better validate the efficiency of the proposed scheme, we also explored the computation time delay to construct the anonymous cloaking region. It can be seen in Fig. 11 that during the process of constructing the cloaking region, the required time delay of our scheme increases with the increment of k. When the privacy protection requirement is k=20, the time delay required is only 0.63s. In particular, the average growth rate of the time delay is lower when the construction of the anonymity domain requirement k is below 14.

Conclusion

In this article, we propose a trusted de-swinging k-anonymity scheme in view of the behavior deception and service swings. In particular, to curb malicious swinging behaviors of nodes, we design a de-swinging reputation evaluation method. With this method, a trusted cloaking area is constructed, where both request nodes and assist nodes only cooperate with their reliance. Furthermore, we design the reputation contracts to calculate reputation and store the scores automatically based on smart contracts. Security analysis and simulation results show that our proposed scheme can resist malicious attacks, identify malicious nodes quickly, and encourage users to participate in the construction process of cloaking areas honestly. Specifically, the highest percentage of malicious users detected by the proposed scheme in the 30th round was 97.13% (first case) and 81.13% (second case). Further, the time delay required by this scheme is within an acceptable range. As shown in Fig. 11, the time delay is 0.63s when the privacy protection requirement is k=20.

However, as the value of k increases, its computation time consumption increases almost linearly. When the user is in a sensitive location with high privacy protection requirements, the delay time of this scheme is high. Moreover, the location privacy requirements of the requesting node are inconsistent with the quality of service. There-

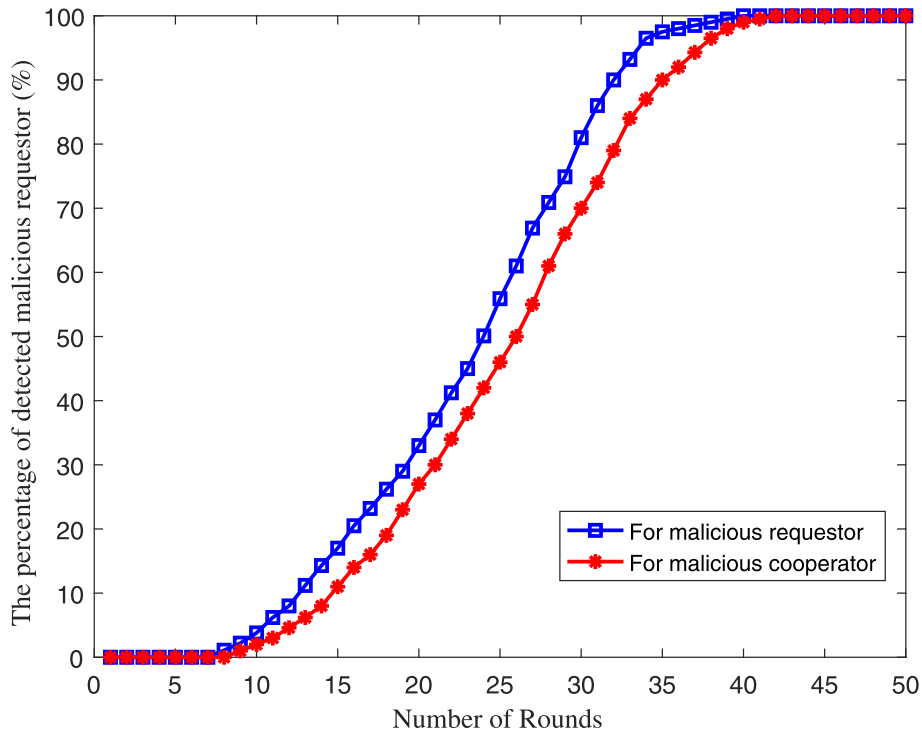

Fig. 10 The percent of case 2 detected malicious nodes

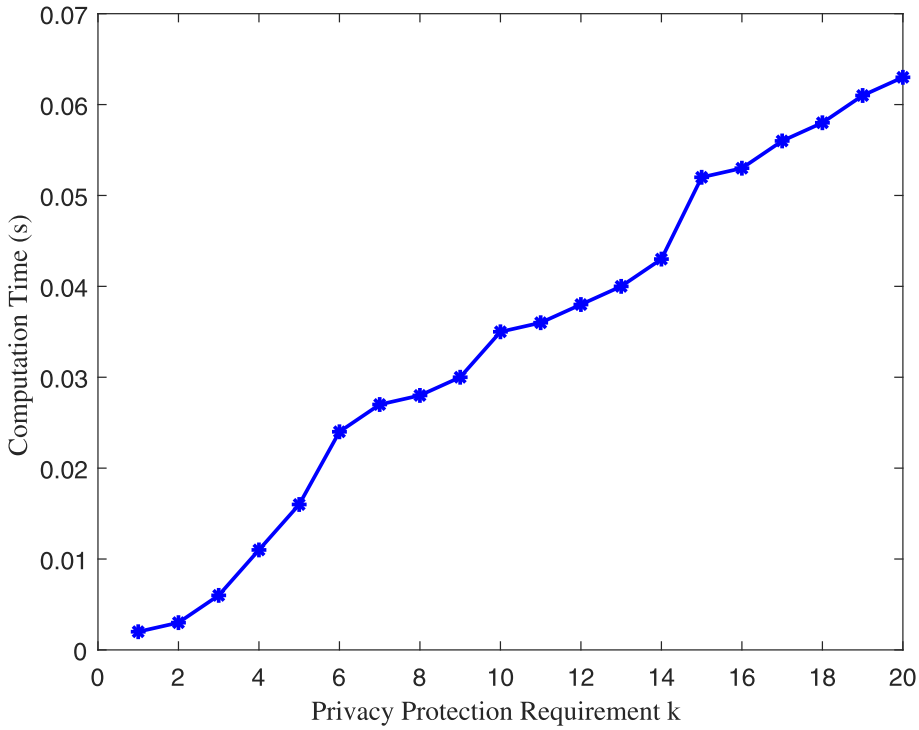

Fig. 11 The computation time of constructing the anonymous cloaking

fore, we will establish a game model by balancing the demand for privacy protection and the quality of service (computational time overhead) to optimize the value of k in our future works.

Abbreviations

DREM: De-swinging reputation evaluation method; LBS: Location-Based service; IoT: Internet of things; ASP: Anonymous service provider

Acknowledgements

This research was supported by both State Key Laboratory of Public Big Data and College of Computer Science and Technology that of GUIzhou University.

Authors' contributions

MY was a major contributor in writing the manuscript as a 1st Author and others were Co-Corresponding Authors. BY, YC and TL proposed some ideas. YY gave some important suggestions for this paper. XQ and XY analyzed the result. All authors read and approved the final manuscript.

Funding

This study is supported by Foundation of National Natural Science Foundation of China (Grant Number: 61962009); Major Scientific and Technological Special Project of Guizhou Province(20183001); Science and Technology Support Plan of Guizhou Province ([2020] 2Y011);Talent project of Guizhou Big Data Academy Guizhou Provincial Key Laboratory of Public Big Data.([2018]01).

Availability of data and materials

The datasets generated and analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Competing interests

The authors declare that they have no competing interests.

Author details

[1]State Key Laboratory of Public Big Data, College of Computer Science and Technology, Guizhou University, Guiyang, China. [2]Information Technology Innovation Service Center of Guizhou Province, National University of Sciences, Guiyang, China. [3]School of Cyberspace Security, Beijing University of Posts and Telecommnuications, Beijing, China. [4]Guizhou CoVision Science & Technology Co., Ltd, Guiyang, China. [5]School of Information Science and Engineering, Shandong Normal University, Jinan, China.

Received: 1 September 2021 Accepted: 5 November 2021
Published online: 08 January 2022

References

1. Xuejun ZZW, Xiaolin G (2015) Overview of research on privacy protection of location services. J Softw 26(9):2373–2395
2. Beresford AR, Stajano F (2003) Location privacy in pervasive computing. IEEE Pervasive Comput 2(1):46–55. https://doi.org/10.1109/MPRV.2003.1186725
3. Liu Y, Pei A, Wang F, Yang Y, Zhang X, Wang H, Dai H, Qi L, Ma R (2021) An attention-based category-aware gru model for the next poi recommendation. Int J Intell Syst 36(7):3174–3189. https://doi.org/10.1002/int.22412
4. Nitu P, Coelho J, Madiraju P (2021) Improvising personalized travel recommendation system with recency effects. Big Data Mining Analytics 4(3):139–154. https://doi.org/10.26599/BDMA.2020.9020026
5. Jiang H, Li J, Zhao P, Zeng F, Xiao Z, Iyengar A (2021) Location privacy-preserving mechanisms in location-based services: A comprehensive survey. ACM Comput Surv 54(1). https://doi.org/10.1145/3423165
6. Khazbak Y, Fan J, Zhu S, Cao G (2020) Preserving personalized location privacy in ride-hailing service. Tsinghua Sci Technol 25(6):743–757. https://doi.org/10.26599/TST.2020.9010010
7. He T, Ciftcioglu EN, Wang S, Chan KS (2017) Location privacy in mobile edge clouds: A chaff-based approach. IEEE J Sel Areas Commun 35(11):2625–2636. https://doi.org/10.1109/JSAC.2017.2760179
8. Ma C, Yan Z, Chen CW (2019) ***sspa***—lbs: Scalable and social-friendly privacy-aware location-based services. IEEE Trans Multimed 21(8):2146–2156. https://doi.org/10.1109/TMM.2019.2892300

9. Li J, Zeng F, Xiao Z, Jiang H, Zheng Z, Liu W, Ren J (2020) Drive2friends: Inferring social relationships from individual vehicle mobility data. IEEE Internet Things J 7(6):5116–5127. https://doi.org/10.1109/JIOT.2020.2974669
10. Gruteser M, Grunwald D (2003) Anonymous usage of location-based services through spatial and temporal cloaking. In: Proceedings of the $1st$ International Conference on Mobile Systems, Applications and Services (MobiSys '03). Association for Computing Machinery, New York. pp 31–42. https://doi.org/10.1145/1066116.1189037
11. Chow C-Y, Mokbel MF, Liu X (2006) A peer-to-peer spatial cloaking algorithm for anonymous location-based service. In: Proceedings of the 14th Annual ACM International Symposium on Advances in Geographic Information Systems (GIS '06). Association for Computing Machinery, New York. pp 171–178. https://doi.org/10.1145/1183471.1183500
12. Sun G, Liao D, Li H, Yu H, Chang V (2017) L2p2: A location-label based approach for privacy preserving in lbs. Futur Gener Comput Syst 74:375–384. https://doi.org/10.1016/j.future.2016.08.023
13. Zhong G, Hengartner U (2009) A distributed k-anonymity protocol for location privacy. In: 2009 IEEE International Conference on Pervasive Computing and Communications. pp 1–10. https://doi.org/10.1109/PERCOM.2009.4912774
14. Ghaffari M, Ghadiri N, Manshaei MH, Lahijani MS (2017) $p4qs$: A peer-to-peer privacy preserving query service for location-based mobile applications. IEEE Trans Veh Technol 66(10):9458–9469. https://doi.org/10.1109/TVT.2017.2703631
15. Zhao P, Li J, Zeng F, Xiao F, Wang C, Jiang H (2018) Illia: Enabling k -anonymity-based privacy preserving against location injection attacks in continuous lbs queries. IEEE Internet Things J 5(2):1033–1042. https://doi.org/10.1109/JIOT.2018.2799545
16. Ying B, Nayak A (2019) A distributed social-aware location protection method in untrusted vehicular social networks. IEEE Trans Veh Technol 68(6):6114–6124. https://doi.org/10.1109/TVT.2019.2906819
17. Li Z, Li W, Wen Q, Chen J, Yin W, Liang K (2019) An efficient blind filter: Location privacy protection and the access control in fintech. Futur Gener Comput Syst 100:797–810. https://doi.org/10.1016/j.future.2019.04.026
18. Wang J, Cai Z, Yu J (2020) Achieving personalized k-anonymity-based content privacy for autonomous vehicles in cps. IEEE Trans Ind Inform 16(6):4242–4251. https://doi.org/10.1109/TII.2019.2950057
19. Li F, Ge R, Zhou H, Wang Y, Liu Z, Yu X Tesia: A trusted efficient service evaluation model in internet of things based on improved aggregation signature. Concurrency Comput Pract Experience:5739. https://doi.org/10.1002/cpe.5739
20. Wang Y, Yang G, Li T, Li F, Tian Y, Yu X (2020) Belief and fairness: A secure two — party protocol toward the view of entropy for iot devices. J Netw Comput Appl 161:102641. https://doi.org/10.1016/j.jnca.2020.102641
21. Y S, Z K, Z H, Z G K-anonymous location privacy protections scheme based on the integration of game theory and blockchain. Comput Appl Res 38(5):7. https://doi.org/10.19734/j.issn.1001-3695.2019.10.0654
22. Liu H, Xing-Hua LI, Luo B, Wang YW, Ren YB, Jian-Feng MA, Ding HF, Information SO (2019) Distributed k-anonymity location privacy protection scheme based on blockchain. J Comput
23. Li T, Wang Z, Yang G, Cui Y, Chen Y, Yu X (2021) Semi-selfish mining based on hidden markov decision process. Int J Intell Syst 36(7):3596–3612. https://doi.org/10.1002/int.22428
24. Li T, Chen Y, Wang Y, Wang Y, Zhao M, Zhu H, Tian Y, Yu X, Yang Y (2020) Rational protocols and attacks in blockchain system. Secur Commun Netw. https://doi.org/10.1155/2020/8839047
25. Wang Y, Yang G, Bracciali A, Leung H.-f., Tian H, Ke L, Yu X (2020) Incentive compatible and anti — compounding of wealth in proof-of-stake. Inf Sci 530:85–94. https://doi.org/10.1016/j.ins.2020.03.098
26. Lu Z, Liu W, Wang Q, Qu G, Liu Z (2018) A privacy — preserving trust model based on blockchain for vanets. IEEE Access 6:45655–45664. https://doi.org/10.1109/ACCESS.2018.2864189
27. Li X, Miao M, Liu H, Ma J, Li K-C (2017) An incentive mechanism for k-anonymity in lbs privacy protection based on credit mechanism. Soft Comput 21. https://doi.org/10.1007/s00500-016-2040-2
28. Yang D, Fang X, Xue G (2013) Truthful incentive mechanisms for k-anonymity location privacy. In: 2013 Proceedings IEEE INFOCOM. pp 2994–3002. https://doi.org/10.1109/INFCOM.2013.6567111
29. Li F, Wang D, Wang Y, Yu X, Wu N, Yu J, Zhou H (2020) Wireless communications and mobile computing blockchain-based trust management in distributed internet of things. Wirel Commun Mob Comput 2020:1–12. https://doi.org/10.1155/2020/8864533
30. Li B, Liang R, Zhu D, Chen W, Lin Q (2021) Blockchain-based trust management model for location privacy preserving in vanet. IEEE Trans Intell Transp Syst 22(6):3765–3775. https://doi.org/10.1109/TITS.2020.3035869
31. Zhang Y, Tong W, Zhong S (2016) On designing satisfaction-ratio-aware truthful incentive mechanisms for k -anonymity location privacy. Trans Info For Sec 11(11):2528–2541. https://doi.org/10.1109/TIFS.2016.2587241
32. Li X, Miao M, Liu H, Ma J, Li K-C (2017) An incentive mechanism for k — anonymity in lbs privacy protection based on credit mechanism. Soft Comput 21. https://doi.org/10.1007/s00500-016-2040-2
33. Chow CY, F. MM (2007) Enabling private continuous queries for revealed user locations. In: Advances in Spatial and Temporal Databases, 10th International Symposium. pp 16–18. https://doi.org/10.1007/978-3-540-73540-3-15
34. Gong X, Chen X, Xing K, Shin D-H, Zhang M, Zhang J (2015) Personalized location privacy in mobile networks: A social group utility approach. In: 2015 IEEE Conference on Computer Communications (INFOCOM). pp 1008–1016. https://doi.org/10.1109/INFOCOM.2015.7218473
35. Yuan D, Li Q, Li G, Wang Q, Ren K (2020) Priradar: A privacy — preserving framework for spatial crowdsourcing. IEEE Trans Inf Forensic Secur 15:299–314. https://doi.org/10.1109/TIFS.2019.2913232
36. Qi L, Hu C, Zhang X, Khosravi MR, Sharma S, Pang S, Wang T (2021) Privacy — aware data fusion and prediction with spatial — temporal context for smart city industrial environment. IEEE Trans Ind Inform 17(6):4159–4167. https://doi.org/10.1109/TII.2020.3012157
37. Kumari R, Kumar S, Poonia RC, Singh V, Raja L, Bhatnagar V, Agarwal P (2021) Analysis and predictions of spread, recovery, and death caused by covid —19 in india. Big Data Min Analytics 4(2):65–75. https://doi.org/10.26599/BDMA.2020.9020013
38. Xu S, Chen X, He Y (2021) Evchain: An anonymous blockchain-based system for charging-connected electric vehicles. Tsinghua Sci Technol 26(6):845–856. https://doi.org/10.26599/TST.2020.9010043
39. Chen Y, Sun J, Yang Y, Li T, Niu X, Zhou H Pssp: A source location privacy protection scheme based on sector phantom routing in wsns. Int J Intell Syst. https://doi.org/10.1002/int.22666
40. Azrour M, Mabrouki J, Guezzaz A, Farhaoui Y (2021) New enhanced authentication protocol for internet of things. Big Data Min Analytics 4(1):1–9. https://doi.org/10.26599/BDMA.2020.9020010
41. Mabrouki J, Azrour M, Dhiba D, Farhaoui Y, Hajjaji SE (2021) Iot-based data logger for weather monitoring using arduino-based wireless sensor networks with remote graphical application and alerts. Big Data Min Analytics 4(1):25–32. https://doi.org/10.26599/BDMA.2020.9020018
42. Chen H, Zhang Y, Cao Y, Xie J (2021) Security issues and defensive approaches in deep learning frameworks. Tsinghua Sci Technol 26(6):894–905. https://doi.org/10.26599/TST.2020.9010050
43. Li J, Pei X, Wang X, Yao D, Zhang Y, Yue Y (2021) Transportation mode identification with gps trajectory data and gis information. Tsinghua Sci Technol 26(4):403–416. https://doi.org/10.26599/TST.2020.9010014

Publisher's Note