Toward Better Data Veracity in Mobile Cloud Computing: A Context-Aware and Incentive-Based Reputation Mechanism

Hui Lin[1], Jia Hu[2*], Youliang Tian[3], Li Yang[4], Li Xu[1]

[1]Fujian Provincial Key Laboratory of Network Security and Cryptology, School of Mathematics and Computer Science, Fujian Normal University, Fuzhou, 350007 China.
[2]College of Engineering, Mathematics and Physical Sciences, University of Exeter, Exeter, EX4 4QF, UK.
[3] Guizhou Provincial Key Laboratory of Public Big Data, College of Computer Science & Technology, Guizhou University, Guiyang 550025, China.
[4] School of Cyber Engineering, Xidian University, Xi'an 710071, China.

Abstract: As a promising next-generation computing paradigm, Mobile Cloud Computing (MCC) enables the large-scale collection and big data processing of personal private data. An important but often overlooked V of big data is data veracity, which ensures that the data used are trusted, authentic, accurate and protected from unauthorized access and modification. In order to realize the veracity of data in MCC, specific trust models and approaches must be developed. In this paper, a Category-based Context-aware and Recommendation incentive-based reputation Mechanism (CCRM) is proposed to defend against internal attacks and enhance data veracity in MCC. In the CCRM, innovative methods, including a data category and context sensing technology, a security relevance evaluation model, and a Vickrey-Clark-Groves (VCG)-based recommendation incentive scheme, are integrated into the process of reputation evaluation. Cost analysis indicates that the CCRM has a linear communication and computation complexity. Simulation results demonstrate the superior performance of the CCRM compared to existing reputation mechanisms under internal collusion attacks and bad mouthing attacks.

Keywords: data veracity; mobile cloud computing; reputation mechanism; mechanism design

1. Introduction

Mobile Cloud Computing (MCC) combines cloud computing and mobile computing to provide mobile users with data storage and processing services in clouds, such as Amazon, Google AppEngine and Microsoft Azure, that perform resource-intensive computing [31, 32]. MCC is a highly promising technology trend for the future of mobile computing, and for this reason, there has been a phenomenal burst of research activities in MCC. Although MCC has attracted significant research and development efforts, there are salient open issues and challenges in the area of security and trust in MCC, which is an essential factor for the

success of the burgeoning MCC paradigm [1, 26, 27, 31, 32].

MCC enables the large-scale collection and processing of personal private data such as individuals' locations and electronic medical records [5, 6, 26, 32]. The processing of this information using big data analytics has become a hot topic in MCC. In order to avoid making decisions based on the analysis of uncertain and imprecise data, it is crucial to maintain a high level of data veracity, which is often overlooked. But it is just as important as the other three V's of Big Data: Volume, Velocity and Variety. Data veracity includes two aspects: data certainty defined by their statistical reliability; and data trustworthiness defined by a number of factors including data origin, collection and processing methods, such as trusted infrastructure and facility [29]. Thus, apart from data confidentiality and privacy, data provenance must be certified and data must be accurate, complete and up-do-date as well [3, 25]. Since ubiquitous access to the Internet in MCC exposes critical data and privacy information to new security threats, a number of research works have been focused on security and trust to cope with these new threats and enhance the data veracity in MCC.

Data veracity shows how much the data used are trusted, authentic and protected from unauthorized access and modification. There are many security challenges in data veracity such as external denial-of-service, credential stealing, remote code injection, data integrity attacks, internal attacks, and supply chain attacks [10]. Consequently, the availability, confidentiality, and integrity of both the original data and the data analytics results are threatened by these attacks, e.g., the degraded availability of a big data system, the compromised confidentiality of the data and analytics, and the violated integrity of the data and analytic results.

As an effort to tackle the aforementioned challenges, this paper focuses on the aspect of data trustworthiness to enhance data veracity through designing a reputation mechanism to defend against internal attacks in MCC. A new Category-based Context aware and Recommendation incentive reputation Mechanism (CCRM) is proposed, which incorporates innovative approaches in terms of data categories, context sensing, security relevance and recommendation incentive. To the best of our knowledge, our work is one of the first to describe this front of data veracity in MCC. The major contributions of this work include the following:

(1) This paper proposes a new Category-based Context aware Reputation Mechanism (CCRM) to defend against the internal threats for enhancing data veracity in MCC.

(2) The CCRM incorporates three key innovations: a Vickrey-Clark-Groves (VCG)-based distributed cheat-proof recommendation incentive scheme, a security level-based data category method, and a user context sensing technology.

(3) Extensive OPNET simulation experiments demonstrate that the CCRM improves the performance of the reputation mechanism compared to the state-of-the-art including the RP-CRM [14], ARTSense [23] and Harmony [22] mechanisms. The CCRM can effectively defend against internal collusion attacks and bad mouthing attacks to enhance data veracity in MCC.

The remainder of this paper is organized as follows. Section 2 presents a brief review of related work, Section 3 describes network and adversary models, Section 4 introduces the implementation details of the CCRM, Section 5 analyzes the cost and evaluate the performance of the CCRM. Finally, Section 6 presents the paper's conclusions .

2. Related Work

Data veracity is becoming a research hotspot of big data and there have been many related studies in the literature [2, 4, 10, 16, 20, 15]. For example, Kepner et al. [10] introduced a new technique called Computing on Masked Data (CMD) to improve data veracity while allowing a wide range of computations and queries to be performed with low overhead by combining efficient cryptographic encryption methods with an associative array representation of big data. Bodnar et al. [4] proposed a veracity assessment model for information dissemination on social media networks that combines natural language processing and machine learning algorithms to mine textual content generated by each user. Sanger et al. [20] introduced two veracity research branches emerging from the combination of the terms of interest, namely Big Data for Trust and Trust in Big Data. Aman et al. [2] proposed a two-stage solution for building an electricity consumption prediction model to address the problem of data veracity raised in the context of Smart Electricity Grids. Lukoianova and Rubin [16] focus on veracity as a critical quality factor and introduce a big data veracity index that combines the three dimensions of subjectivity, deception and implausibility.

Data veracity includes two aspects: data certainty and data trustworthiness. This paper focuses on the aspect of data trustworthiness to enhance data veracity using a reputation mechanism to defend against internal attacks in MCC. Tremendous research efforts have been focused on the reputation mechanism as a key scheme for managing trust to improve data security and privacy. Since this paper investigates the data veracity issue in MCC, in the following, we mainly review the existing research results regarding reputation mechanisms in MCC.

Kim et al. [12] proposed a trust management mechanism for reliable data integration, management and applications in MCC. The mechanism suggested a method to quantify a one-dimensional trusting relationship based on the analysis of telephone call data from mobile devices. Liu et al. [15] presented a reputation mechanism to recognize selfish nodes much earlier and reduce the convergence time for isolating selfish nodes by combining familiarity values with subjective opinions. Hammam et al. [8] proposed a trust management system (TMC) for mobile ad-hoc clouds to verify that participants are reliable, available, and harmless. The TMC considered availability, neighbors' evaluation and response quality, and task completeness; it also calculated the reputation trust value for nodes. Shen et al. [22] developed an integrated reputation management platform, Harmony, for collaborative cloud computing. Harmony incorporates an integrated reputation management component, a multi-QoS-oriented resource selection component and a price-assisted resource control component to enhance their mutual interactions for efficient and trustworthy resource sharing among clouds. Zhang et al. [30] presented a general framework to jointly design incentive mechanisms and reputation schemes in social cloud systems. The proposed framework combined a repeated game framework-based incentive mechanism with a differential reputation-based reward/punishment scheme to incentivize users to contribute their resources.

In the existing research on data veracity, many studies were based on the assumption that the participants are trustworthy, thus ignoring the internal security threats launched by an inside attacker that has a legal identity and gives dishonest recommendations to frame up good parties and/or boost trust values of malicious peers. Meanwhile, most existing reputation-based trust models in MCC were based on the traditional cryptographic encryption and authentication techniques without considering internal security threats. Consequently, it is

an open problem and a challenging task to design a reputation mechanism to prevent internal attacks in order to enhance the data veracity in MCC.

3. Network and Adversary Model

A. Network Model

In this paper, we focus on the network environment of MCC, which is considered to be a viable solution to implement fast, large-scale big data applications [11, 34]. A typical MCC architecture, which consists of a mobile client network, a wireless mesh backbone network and a cloud service platform, is depicted in Fig. 1. The mobile client is connected to the base transceiver station (BTS) and accesses the mesh backbone via the mesh router; these are linked to each other and communicate with the cloud through the Internet. The cloud service platform includes cloud servers to offer data-rich services such as queries of electronic medical records.

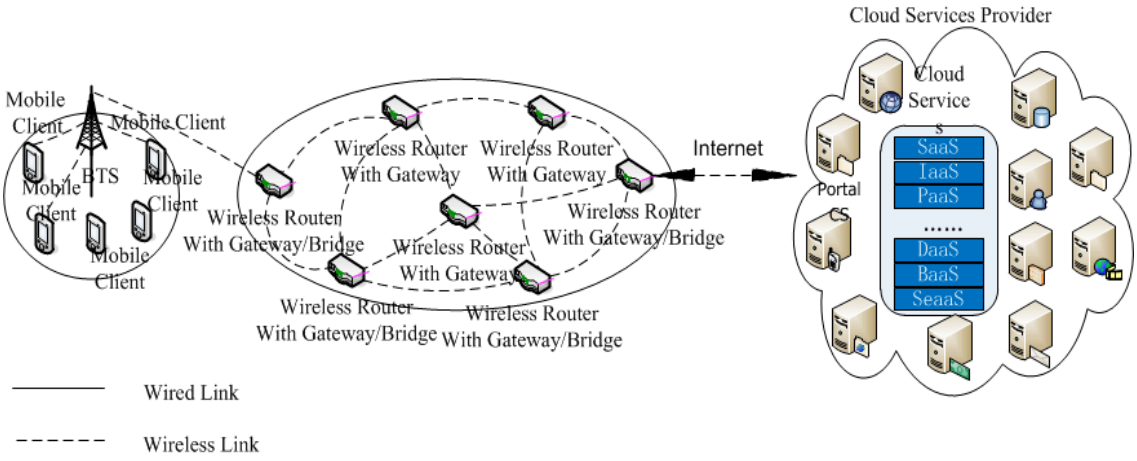

Fig. 1. An Architecture of MCC

B. Adversary Model

This paper focuses on the internal security threats [13, 14] in MCC that can affect data veracity. The internal threats are launched by an inside attacker who is a legal and certified user. The internal attacks may compromise certain users and gain full control of them. Once users are compromised, the attacker can gain access to all stored information, including public keys and private keys [28]. The attacker could also reprogram the captured users to behave in a malicious manner [27]. Therefore, the traditional encryption and authentication techniques

may no longer be effective [33]. The specific internal attacks considered in this paper are as follows:

Collusion attacks: attackers can collude and provide either high or low recommendations to each other.

Bad mouthing attacks: attackers provide dishonest recommendations to frame up good parties and/or boost reputation values of malicious peers.

4. Category-Based Context-Aware Reputation Mechanism (CCRM)

In this section, we elaborate on the proposed Category-based Context-aware Reputation Mechanism (CCRM), which integrates the reputation evaluation with data category [17, 23], context-awareness technologies [19] and the VCG mechanism [7, 18, 24] to defend against the insider threat and enhance the data veracity in MCC. The CCRM is implemented in both mobile clients and cloud service providers to perform bidirectional reputation evaluation. The CCRM includes three phases: direct reputation computation, recommended reputation computation and final reputation computation.

In CCRM, data are classified into different categories based on the sensitivity level of data. The higher is the sensitivity level of data in a category, the higher is the user reputation required for a user to access the data category. The details of the reputation computation are described as follows. The main notations and symbols used in this paper are summarized in Table 1.

Table 1. Main notations and symbols

$R_{y:x}^{\text{Direct}}$, $R_{y:x}^{\text{Rec}}$, $R_{y:x}^{\text{Final}}$	The *x's* direct, recommended and final reputation towards *y*
C, *c*	The set of the data categories and a specific data category
AC_s^c, AC_f^c	The number of successful and unsuccessful accesses to the data category *c*
p_{un}^{dir}	The uncertainty about the history interaction
α_{time}, $\alpha_{location}$, α_c	The weight factors that determine how much the time, location and data category of the interactions affect $R_{y:x}^{\text{Direct}}$
$E_{location}$, E_{time}	The error of location sensing and time sensing
$\beta_{location}$, β_{time}	The location sensitivity and time sensitivity parameters
N_j, N_{slot}	The number of times that *x*'s historical data access category is confirmed as category *j,* the number of time slots
Q,q, γ	The transfer efficiency, bandwidth requirement, signal-to-noise ratio

	of the receiver
$V(q_i)$, $P(q_i)$	The profit function that gives the profits gained by user *i* when it provides the recommendation, the cost of user *i*
p_{cost}	The cost of the bandwidth usage
o, $b_{i,j}$	The set of bandwidth allocation, the allocation that bandwidth *i* is assigned to the user *j*
π^*, $o*$	The system's best profit, the best allocation result
$T_i(q_i, o)$	The taxes for a user *i*
$\sum_{j\neq i} u_j(q_j, o_i^*)$, $\sum_{j\neq i} u_j(q_j, o_{-i})$	The total utility of all the other users when *i* participates and withdraws
$\theta_{(i,y)}$, $\theta_{(i,x)}^{sl}$	The security level relevance factor between *y* and the recommendation user *i*, the security level relevance factor between *i* and the recommended user *x*
η_1, η_2	The weight factors used to determine how much the direct reputation and integrated recommended reputation affect the final reputation
f^{r}, f^{θ}	The reputation and the security relevance fade factors
m^{t_i}, m^{t_n}	The number of the objects within the recommendation users set at times t_i and t_n

4.1. Direct Reputation Computation

The direct reputation computation is run at each user that stores its historical opinion towards the others in the relevant local database. When a user wants to request (or provide) a service from (or to) another user (including unknown users), it will send a request message to all neighboring users. Each neighboring user receiving the request will first execute the direct reputation computation function to evaluate the requestor's direct reputation and judge whether it is a malicious user.

Suppose *x* and *y* are the client and service provider, respectively. The direct reputation of *x* toward *y*, $R_{y:x}^{\mathrm{Direct}}$, can be computed as:

$$R_{y:x}^{\mathrm{Direct}} = \frac{1}{|\mathrm{C}|} * \sum_{c\in\mathbf{C}} [(1 - p_{un}^{dir}) * (AC_s^c / AC_f^c)] \quad (1)$$

where **C** is the set of the data categories and *c* stands for a specific data category. AC_s^c and AC_f^c denote the number of successful or unsuccessful access attempts to the data category *c*, respectively. p_{un}^{dir} denotes the uncertainty about the historical interaction, which is given by:

$$p_{un}^{dir} = (1 - \alpha_{time}) * (1 - \alpha_{location}) * (1 - \alpha_c) \quad (2)$$

where α_{time}, $\alpha_{location}$ and α_c are the weight factors that determine how much the time, location and accessed data category of the interaction affect $R_{y:x}^{\text{Direct}}$, respectively.

In CCRM, a user's location will influence its reputation among the other users, so if a user's location is far away from the expected location then it is not as trustworthy as a user on a nearby location. We denote the expected location and the actual location as *L* and *L'*, respectively. We define |L-L'| as the distance between them and $E_{location}$ as the error of location sensing. We then formally define the location factor $\alpha_{location}$ as:

$$\alpha_{location} = e^{-E_{location} * \beta_{location}} * (1 - e^{-|L-L'| * \beta_{location}}) \tag{3}$$

where $\beta_{location}$ is the parameter that controls the weight of the location factor's influence on reputation.

Time is another critical factor as the historical interaction usually has a great reference value. Similarly to the location factor, we denote T' as the time instant of last interaction between the provider and the requestor and T as the time instant of current interaction between the provider and the requestor. We define E_{time} as the error of time sensing. Next, the time factor α_{time} can be computed as

$$\alpha_{time} = e^{-E_{time} * \beta_{time}} * (1 - e^{-|T-T'| * \beta_{time}}) \tag{4}$$

where β_{time} is the parameter that controls the weight of the time factor's influence on the reputation.

In CCRM, similar to the location and time factors, the historical records of accessed data category will also influence the direct reputation computation. High and low category data are the data category with high data sensitivity and low data sensitivity, respectively. For a user used to access the low category data, its sudden access to a higher category data is noteworthy. Therefore, we define and compute the accessed data category factor α_c as:

$$\begin{cases} \alpha_c = E(\mu_i) \\ \mu_i = \sum_{j=1h}^{|C|} N_j \Big/ \sum_{j=1}^{|C|} N_j \end{cases}, (i = 1 \quad N_{slot}) \tag{5}$$

where μ_i is the rate between the number of accesses to the data categories higher than the threshold *Th* and the total number of accesses to all categories. N_j represents the number of times that *x*'s historical accessed data category is confirmed as category *j*, and N_{slot} denotes the number of the time slots.

The details of the unidirectional direct reputation computation are shown in Algorithm 1.

Algorithm 1: Direct Reputation Computation

Input: Requester *x's* information
Output: Whether *x* is a malicious node or not

1. Begin
2. Requester *x* sends a *Request* message;
3. *x's* neighbor nodes such as *y* receives the *Request* message;
4. If $\left(x's\left|L-L'\right|>TH_{location}^{down}\right)\wedge\left(x's\left|T-T''\right|>TH_{time}^{down}\right)$ then
5. *y* executes the Direct Reputation Computation and returns the result as:
6. R^{Direct}=Direct_reputation (*x)*;
7. Else
8. *y* drops the *Request* message;
9. End if
10. If $(R^{Direct}>TH_{direct}^{upper})$ then
11. $R^{Final}=R^{Direct}$;
12. Else if $(TH_{direct}^{down}<R^{Direct}<TH_{direct}^{upper})$ then
13. *y* executes the Recommendation Reputation Query Function;
14. *y* executes the Recommendation Reputation Computation;
15. *y* executes the Final Reputation Computation and gets the R^{Final};
16. Else
17. $R^{Final}=-1$;
18. End if
19. If $(R^{Final}<TH_{final}^{down})$ then
20. *x* is considered as a malicious node and will be isolated;
21. Else if $(TH_{final}^{down}<R^{Final}<TH_{final}^{upper})$ then
22. *x* will be punished by decreasing its reputation value;
23. Else
24. *x* is considered as a trustworthy node;
25. *y* sends *Accept* message to *x*;
26. End if
27. End

In order to gain the service, requester *x* sends a request message to all neighboring users. Each neighboring user receiving the request first executes the direct reputation computation function to evaluate the requestor's direct reputation and judge whether it is a malicious user. Suppose *y* is one of the service providers and it receives the *Request* message. If both the distance between *x's* actual location and expected location and the gap between the historical interaction time instant and current time instant are greater than the thresholds $TH_{location}^{down}$

and TH_{time}^{down}, y computes the direct reputation; otherwise y drops the *Request* message. If the direct reputation is greater than the threshold TH_{direct}^{upper}, we set the final reputation equal to the direct reputation. If the result of the direct reputation is between the thresholds TH_{direct}^{down} and TH_{direct}^{upper}, then y executes the recommendation reputation query and computation and computes the final reputation. If the result of the direct reputation is less than the TH_{direct}^{down}, we set the final reputation equal to -1. When y finds the final reputation, if it is less than the TH_{final}^{down}, then x is considered as a malicious node and will be isolated. If the final reputation is between the thresholds TH_{final}^{down} and TH_{final}^{upper}, x will be punished by decreasing its reputation value. If the final reputation is greater than the TH_{final}^{upper}, x is considered as a trustworthy node and y will send an *Accept* message to x.

4. 2. Recommended Reputation Computation

If the direct reputation computation in section 4.1 cannot lead to a decision, y will first execute the recommended reputation query using Algorithm 2 to query x's reputation from its neighbors. Afterwards, y will compute the integrated recommended reputation combining the received replies of recommended reputations to the query, which will be described in the subsection 4.2.2.

Algorithm 2: Recommended Reputation Query

Input: Users x and y's information
Output: x's reputation

1. Begin
2. y sends a *Query* message to neighbors;
3. Wait (3-5 seconds);
4. y's neighbor user such as k receives the *Query* message;
5. If $\left(y'security\,level > TH_{sl}^{down}\right)$ then
6. k retrieves its direct reputation opinion about x on local reputation database
7. and returns it as:
8. $R_x^{\text{Direct}} = Reputation\_query(x, Final\,Reputation\,Database)$;
9. k sends *Reply* (y, R_x^{Direct});
10. Else
11. k drops the *Query* message;
12. End if
13. End

Traditional reputation mechanisms improve the trustworthiness of recommendations through weighted summation of recommendations from different recommenders. In an open network environment such as MCC, however, these mechanisms must face the significant problems caused by selfish and malicious users who refuse to render recommendations in

order to avoid consuming limited resources or provide dishonest recommendations to launch collusion or bad mouthing attacks.

To overcome the above shortcomings, in this subsection, we first propose a VCG based Distributed Cheat-proof Recommendation incentive scheme (VDCR). Next, the VDCR is incorporated into the recommended reputation computation process to motivate users to provide honest recommendations.

4.2.1. VDCR Scheme

The VCG mechanism is a dominant strategy mechanism, which belongs to a category of mechanism design. The VCG mechanism can achieve ex-post incentive compatibility (truth-telling is a dominant strategy for every player in the game) [7, 18, 24]. The VCG mechanism has the following attributes:

The mechanism is incentive compatible.

The mechanism is individual rational.

Since users must consume bandwidth resources to provide the recommendation information, the recommendation processes can be naturally modelled as a VCG-based bandwidth auctions process with the user's profit model and system's profit model defined as below.

1) Profit Model of User

Suppose user i wants to provide a recommendation and its bandwidth requirement is q, which is often private information known only to the user. Using the classical transmission model [9], q can be given by:

$$\begin{cases} q = \log_2(1 + Q \cdot \gamma), \\ Q = \dfrac{1.5}{\ln(0.2 / BER^{tar})} \end{cases} \tag{6}$$

where Q is the transfer efficiency and γ is the signal-to-noise ratio of the receiver.

The profit of the user $i, u_i(q_i)$, can be expressed as:

$$u_i(q_i) = \begin{cases} V(q_i) - P(q_i), & purchase \\ 0, & don't\ purchase \end{cases} \tag{7}$$

where $V(q_i)$ is the profit function that gives the profits gained by user i when it provides the recommendation. $V(q_i)$ is given by:

$$
\begin{aligned}
&V(q_i) = aq_i^2 + bq_i \\
&s.t. \\
&\quad q_i = 0, V(0) = 0 \\
&\quad V'(q_i) > 0,\ V''(q_i) < 0
\end{aligned}
\tag{8}
$$

in which a, b are the weight factors. $P(q_i)$ is the cost of user i and can be computed as:

$$P(q_i) = p_{cost} q_i \tag{9}$$

where p_{cost} denotes the cost of the bandwidth usage, which is relevant to the reputation. The higher the reputation, the lower the cost.

Hence, when $q_i > 0$, the profit of the user $i, u_i(q_i)$, is given by:

$$u_i(q_i) = aq_i^2 + (b - p_{cost})q_i \tag{10}$$

2) Profit Model of System

We consider the MCC environment consisting of n users. Let $o = (b_{i,j} \mid i = 1,2, \quad ,m; j = 1,2, \quad ,n)$ denote the set of bandwidth allocation, where $b_{i,j}$ indicates that bandwidth i is assigned to the user j. Next, the system's best profit π^* can be expressed as:

$$\pi^* = \sum_{i=1}^{n} u_i(q_i, o^*) = \max_{o_k \in O} \sum_{i=1}^{n} u_i(q_i, o_k) \tag{11}$$

where o^* is the best allocation result and is given by:

$$o^* = \arg\max_{o_k \in O} \sum_{i=1}^{n} u_i(q_i, o_k) \tag{12}$$

3) VDCR

Based on the above analysis, we can describe the details of the proposed VCG-based distributed cheat-proof recommendation incentive scheme (VDCR).

In the VDCR, user must pay taxes $T(q,o) > 0$ in addition to the cost of bandwidth. The taxes for a user i are denoted as $T_i(q_i, o)$, which is given by

$$T_i(q_i, o^*) = \sum_{j \neq i} u_j(q_j, o_{-i}) - \sum_{j \neq i} u_j(q_j, o^*) \tag{13}$$

where $\sum_{j \neq i} u_j(q_j, o_i^*)$ is the total utility of all other users when i participates. As opposed to $\sum_{j \neq i} u_j(q_j, o_i^*)$, $\sum_{j \neq i} u_j(q_j, o_{-i})$ is the total utility of all other users when i withdraws. Therefore, the best utility of user i can be expressed as:

$$u_i^*(q_i, o^*) = u_i(q_i, o^*) - [\sum_{j \neq i} u_j(q_j, o_{-i}) - \sum_{j \neq i} u_j(q_j, o^*)] \quad (14)$$

Next, we prove that the proposed VDCR is a VCG mechanism.

Theorem 1. The mechanism is incentive compatible.

Proof: suppose user *i* needs q_i units of bandwidth to provide recommendation, but the user applies for $\hat{q}_i$ units and declares that $u_i(\hat{q}_i) > u_i(q_i)$, which returns the outcome of $\hat{o}$. According to the above description, user *i* must pay the taxes as:

$$T_i(q_i, \hat{o}) = \sum_{j \neq i} u_j(q_j, o_{-i}) - \sum_{j \neq i} u_j(q_j, \hat{o})$$

Hence, the final utility of user *i* becomes:

$$\begin{aligned} \hat{u}_i(q_i, \hat{o}) &= u_i(q_i, \hat{o}) - [\sum_{j \neq i} u_j(q_j, o_{-i}) - \sum_{j \neq i} u_j(q_j, \hat{o})] \\ &= u_i(q_i, \hat{o}) + \sum_{j \neq i} u_j(q_j, \hat{o}) - \sum_{j \neq i} u_j(q_j, o_{-i}) \\ &= \pi(\hat{o}) - \sum_{j \neq i} u_j(q_j, o_{-i}) \end{aligned}$$

For $\pi(\hat{o}) \leq \pi^*$, we obtain $\hat{u}_i(q_i, \hat{o}) \leq u_i^*(q_i, o^*)$, which leaves user *i* with no motivation to provide false information; therefore, truth-telling is the best strategy.

Theorem 2. The mechanism is individual rational.

Proof: In an individual rational (IR) mechanism, rational users are expected to gain a higher utility from actively participating in the mechanism than from avoiding it. In the VDCR, we consider the following two malicious behaviors:

(1) The user does not have the relative recommendation information, but it still applies for bandwidth to provide the recommendation.

(2) The user does not have enough to pay the costs and taxes, but it still applies for bandwidth to provide the recommendation.

The utility of the user with these malicious behaviors, $\hat{u}_i(\hat{q}_i, \hat{o})$, can be given by:

$$\hat{u}_i(\hat{q}_i, \hat{o}) = u_i(\hat{q}_i, \hat{o}) - T_i(\hat{q}_i, \hat{o})$$

It is straightforward to see that in both cases $\hat{u}_i(\hat{q}_i, \hat{o}) < 0$, when the incentive compatibility is achieved. Therefore, for user *i*, utility > 0 and participation in the recommendation is an optimal choice, which means that the mechanism is individual rational.

In sum, according to the definition of VCG, the proposed VDCR is a VCG mechanism.

4.2.2. Computation of Integrated Recommended Reputation

Let *sl* stand for security level, which means the security level of the data category that a user can access with its reputation. Suppose *y* receives *n* (*n*>1) recommended reputations, then the integrated recommended reputation, $R_{y:x}^{\text{Rec}}$, can be computed as follows:

(1) Consider there are only two recommenders, *k* and *k'*, and their recommended opinions are in conflict. We say that *k* is more trustworthy than *k'* and $R_{y:x}^{\text{Rec}} = R_{k:x}^{\text{Direct}}$, if any of the following conditions hold:

$$\begin{cases} \theta_{(k,y)} > \theta_{(k',y)} \\ \theta_{(k,y)} = \theta_{(k',y)} \wedge R_{y:k}^{\text{Direct}} > R_{y:k'}^{\text{Direct}} \end{cases} \tag{15}$$

(2) Consider that if there are more than two recommenders, then the $R_{y:x}^{\text{Rec}}$ can be given by:

$$R_{y:x}^{\text{Rec}} = \frac{1}{n} * \sum_{j=1}^{n} \left(\theta_{(j,y)} * R_{j:x}^{\text{Direct}} \right) \tag{16}$$

In Eqs. (15) and (16), $\theta_{(i,y)}$ denotes the security relevance factor between service provider *y* and the recommendation user *i*, which characterizes the difference between the security level of *y* and that of *i*. Suppose user *i* recommends *x* to *y*, we define $\theta_{(i,x)}^{sl}$ as the security relevance factor between the users *i* and *x*. $\theta_{(i,y)}$ can be calculated as:

$$\theta_{(i,y)} = \begin{cases} sl_i < sl_y, \beta_{sl}^1 * \lg(sl_y / (sl_y - sl_i)) + \beta_{sl}^2 * \theta_{(i,x)}^{sl} \\ sl_i \geq sl_y, \ \beta_{sl}^1 * \lg(sl_i - sl_y) + \beta_{sl}^2 * \theta_{(i,x)}^{sl} \\ \beta_{sl}^1 + \beta_{sl}^2 = 1, \ \beta_{sl}^1, \beta_{sl}^2 \in [0,1] \end{cases} \tag{17}$$

where β_{sl}^1 and β_{sl}^2 are the weight factors associated with the numerical difference between security levels of *y* and *i* and the security relevance factor $\theta_{(i,x)}^{sl}$, respectively. $\theta_{(i,x)}^{sl}$ can be given by:

$$\theta_{(i,x)}^{sl} = \begin{cases} sl_i = sl_x, & 1 - \dfrac{\left| R_{y:i} - R_{y:x} \right|}{Th_{sl(i,x)}^{upper} - Th_{sl(i,x)}^{dowm}} \\ sl_i > sl_x, & 1 - sl_x / sl_i \\ sl_i < sl_x, & sl_i / sl_x \end{cases} \tag{18}$$

where $R_{y:i}$ and $R_{y:x}$ are the reputation values of *i* and *x*, respectively.

Algorithm 3 gives the details of the computation of integrated recommended reputation.

Algorithm 3: Computation of Integrated Recommended Reputation

---

Input: *N* recommendation information
Output: Integrated recommended reputation value
1. Begin
2. *y* receives *n-1 Reply* messages with the recommended reputation about *x*
3. and the information of the recommenders;
4. *y* executes the recommenders' selection process;
5. for *i*=1 to *n-1*
6. { If $\left(i's \text{ security level} > TH\right)$ then
7. Put *i* into the recommender set R;
8. *y* executes the security relevance factor $\theta_{(i,y)}$ and $\theta_{(i,x)}^{sl}$ computation;
9. Else
10. *y* drops the *Reply* message;
11. End if }
12. *y* executes the recommenders' selection process again;
13. for *j*=1 to |R|
14. { If $(\theta_{(j:y)} > TH')$ then
15. Put *j* into the recommender set R';
16. End if }
17. *y* executes the recommended reputation computation with R'
18. and returns the result as:
19. R^{Rec}=Rec_reputation (*x)*;
20. End

When *y* receives *n-1 Reply* messages, it first builds a recommender set **R** by comparing each recommender's security level (*sl*) with a threshold, one by one. If a recommender's *sl* is greater than the threshold, *y* will put it into set **R**, and then compute the corresponding relevance factor θ and θ^{sl}. Then, *y* builds a more reliable recommender set **R'** by selecting the recommender from **R** and comparing its security relevance factor θ with the threshold. If a recommender's θ is greater than the threshold, *y* will put it into the new recommender set **R'**. Finally, *y* computes the recommended reputation with **R'** and returns the result.

4.3. Final Reputation Computation

After finding the direct and recommended reputation, the final reputation $R_{y:x}^{\text{Final}}$ can be computed as:

$$\begin{cases} R_{y:x}^{\text{Final}} = \eta_1 * R_{y:x}^{\text{Direct}} + \eta_2 * R_{y:x}^{\text{Rec}} \\ \eta_1 + \eta_2 = 1, \eta_1, \eta_2 \in [0,1] \end{cases} \tag{19}$$

where η_1, η_2 are the weight factors for the direct reputation and integrated recommended reputation, respectively.

4.4. Update of Reputation

Because users' reputations change over time, the direct reputation used at present cannot directly make use of the data stored in the local reputation database without considering the influence of time on the reputation. Denote m^{t_i} and m^{t_n} as the numbers of the objects within the recommendation users set at time t_i and t_n, respectively. Let f^{r} and f^{θ} represent the reputation and the security relevance fade factors, respectively. Next, the direct reputation at time t_n can be updated as below:

$$R_{t_n}^{\text{Direct}} = f^{\mathrm{r}} * f^{\theta} * R_{t_i}^{\text{Final}} \tag{20}$$

$$\begin{cases} f^{\mathrm{r}} = e^{-((R_{t_i}^{\text{Final}})^{-1} * \Delta t)^{2k}} \\ f^{\theta} = \dfrac{\overline{\theta_{(i,y)}^{t_n}} - \overline{\theta_{(i,y)}^{t_i}}}{\overline{\theta_{(i,y)}^{t_n}}} \\ \Delta t = t_n - t_i \end{cases} \tag{21}$$

$$\begin{cases} \overline{\theta_{(i,y)}^{t_i}} = \dfrac{1}{m^{t_i}} \sum_{t=1}^{m^{t_i}} \theta_{(i,y)}^{t_i} \\ \overline{\theta_{(i,y)}^{t_n}} = \dfrac{1}{m^{t_n}} \sum_{t=1}^{m^{t_n}} \theta_{(i,y)}^{t_n} \end{cases} \tag{22}$$

5. Cost Analysis and Performance Evaluation

In the section, we first elaborate on the communication cost and the computation complexity, and then present the performance evaluation of the proposed reputation mechanism CCRM.

5.1. Computation and Communication Cost

The communication cost of CCRM can be calculated as follows:

$$
\begin{aligned}
Cost^{\text{communication}} &= cost_{\text{request}} + cost_{\text{query}} + cost_{\text{reply}}, \\
&= O(n-1) + O(n-1) + O(n-1), \\
&= O(3n-3).
\end{aligned} \tag{23}
$$

where $cost_{\text{request}}$, $cost_{\text{query}}$ and $cost_{\text{reply}}$ represent the number of the request, query, and reply messages transferred in the communication, respectively. n denotes the total number of users.

The computation complexity can be analyzed as below:

$$
\begin{aligned}
Cost^{\text{computation}} &= cost_{\text{direct}} + cost_{\text{vdcr}} + cost_{\text{rec}} + cost_{\text{final}} + cost_{\text{update}}, \\
&= O(n_{\text{direct}}) + O(n_{\text{vdcr}}) + O(n_{\text{rec}}) + O(1) + O(n_{\text{update}}), \\
&= O(4n).
\end{aligned} \tag{24}
$$

where n_{direct}, n_{rec} and n_{update} represent the number of users involved in computing the direct reputation, recommended reputation and updated reputation, respectively, and n_{vdcr} is the number of users involved in running the VDCR mechanism.

The above analysis indicates that both the communication cost and the computation complexity of the CCRM is $O(n)$. Therefore, with the strong computing power of modern mobile devices and the good capacity of contemporary communication networks, the influences of the computation and communication cost are little and thus negligible considering the great benefits of enhanced security and data veracity the proposed mechanism brings to MCC.

5.2. Performance Evaluation

The OPNET [21] simulation experiments were conducted to evaluate the performance of the CCRM in MCC. The simulation scenario includes a mobile client network, a wireless mesh backbone network and a cloud service platform, as shown in Fig. 1. The mobile client network consists of 100 mobile clients. The mesh backbone network has 10 mesh routers and there are 10 service providers in the cloud. The physical layer uses a fixed range transmission model where two nodes can directly communicate with each other only if they are within a certain range, i.e., within a hop. The arrival traffic at each mobile client follows a Constant Bit Rate (CBR) of 1 Mbps and a packet size of 1024 bytes. The channel data rate is 10 Mbps. The Hybrid Wireless Mesh Protocol (HWMP) [13] is used as the routing protocol. The security parameters β_{sl}^{1}, β_{sl}^{2}, η_1, η_2, a, b are 0.5, 0.5, 0.4, 0.6, -10, 40, which are empirical values

obtained from multiple experiments. Of which, β_{sl}^{1} and β_{sl}^{2} are the weight factors in Eq. (17) associated with the numerical difference between security levels of *y* and *i* and the security relevance factor $\theta_{(i,x)}^{sl}$, respectively. η_1 and η_2 are the weight factors in Eq. (19) used to determine how much the direct reputation and integrated reputation affect the final reputation, respectively. *a* and *b* are the weight factors in Eq. (8) to determine the value of the profit function that gives the profits gained by a user when it provides the recommendation. Each data point depicted in the following figures is the average of the results obtained from 100 runs of simulation experiments with a simulation time of 100 s each.

We adopt the MCC-based query system for electronic medical records as the simulation scenario. In the scenario, a mobile client requesting electronic medical records first connects to a BTS, which accesses the mesh backbone via the mesh router. Then, the MC will send a service request to the cloud service provider (CSP) to ask for the electronic medical records. When the CSP receives the request, it will evaluate the trustworthiness of the MC and decide whether to approve the request. The detailed process is described as follows.

First, if there are one or more BTSs that the MC can directly access, it will broadcast a query message to its neighbor clients to query which BTS is more secure and reliable. If there is no BTS that the MC can directly access, it will evaluate the reputation of the neighbor clients based on the CCRM and select a most trustworthy client as a relay node to connect to the BTS. Second, when the request message is transported in the WMN, the CCRM is used to select a trustworthy route in order to prevent the privacy information in the request message from being eavesdropped on or tampered. Third, when the CSP receives the request, it will evaluate the trustworthiness of the MC using the CCRM and decide whether to approve the request.

In this subsection, the performance of the proposed CCRM is compared to the Harmony [22], RP-CRM [14] and ARTSense [23] because the RP-CRM was a similar reputation mechanism proposed in our previous work, while Harmony and ARTSense are the latest proposed related mechanisms. The following performance metrics are evaluated when internal collusion attacks and bad mouthing attacks are present.

Utility of the recommender: The utility obtained by recommenders when they provide

recommendations.

Reputation and its update accuracy: The reputation of a user and the update accuracy of its reputation.

Impact of context information: The impact of contextual information on users' reputations.

Effective recommendation rate (ERR): The ratio of the number of accurate recommendations and the number of all recommendations.

Malicious user detection rate (MDR): The accuracy of detecting and identifying malicious users.

5.2.1 Utility of the Recommender

First, we investigate how the CCRM performs in an honest network and a hostile network, respectively. In the honest network, all the recommenders are normal users, while in the hostile network, the recommenders may be malicious users who give false information with the probability λ.

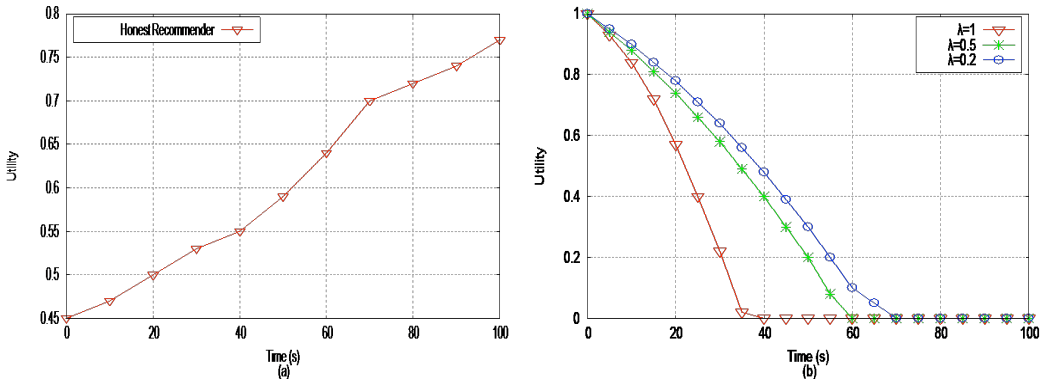

Fig. 2. Utility of the Recommender: (a) Utility of the honest one (b) Utility of the malicious one

The average utility of the recommender in the honest network is shown in Fig. 2 (a). The results show that the more truthful information the recommender provides, the larger average utility it achieves. In the CCRM, the VCG-based VDCR scheme will be run before users provide their recommendations. According to the characteristics of VCG, a user that participates the recommendation and does truth-telling is the best strategy; thus, a normal user in the honest work will always provide truthful information, which leads to an increase in its utility. Moreover, the utility a user gained in a recommendation will be used to update the user's reputation, which leads to the continuous growth of its utility.

In Fig. 2 (b), we analyze how the false recommendation from an adversary would impact its utility. We set three adversaries with the probability of giving false information (λ) being 1, 0.2 and 0.5, respectively. It is assumed that all of them have a utility value of 1 at the beginning of the test. The results show that the average utility of the malicious user is affected by providing dishonest recommendations, and the larger the λ, the faster the utility decreases. When an adversary has a λ of 1 (i.e., always reports false information), its utility drops the fastest to 0. The utility of an adversary who sometimes sends correct information (with λ= 0.2 and 0.5) decreases more slowly. The utility eventually drops to a very low level, however, even if false information is sent with a small probability (λ= 0.2). This is because when the adversary gives a dishonest recommendation, according to the characteristics of the VCG mechanism, the proposed VDCR makes it pay a higher cost, which leads to a decline in its utility. In our mechanism, when the user's utility becomes less than 0, it is reset to 0.

5.2.2 Reputation and Its Update Accuracy

Next, we analyze the impact of the adversary ratio on users' reputations and compare the reputation and its update accuracy of the CCRM with those of the RP-CRM, ARTSence and Harmony. The λ of all adversaries is set to be 1, which represents the worst case scenario.

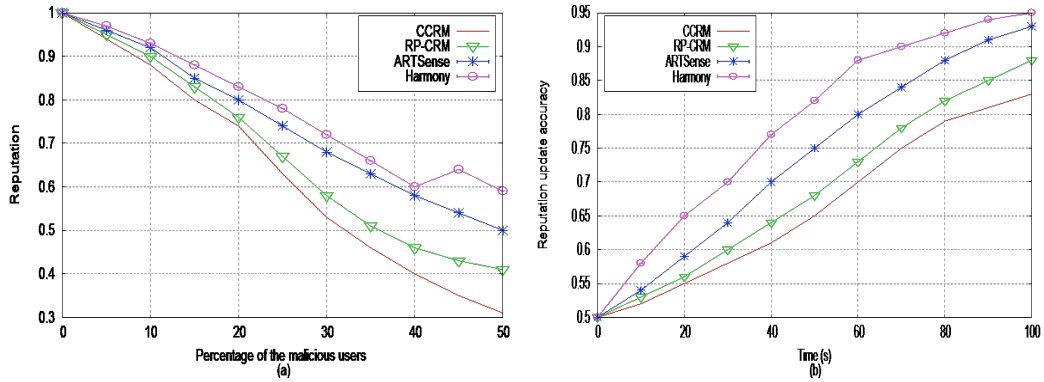

Fig. 3. Reputation and its update accuracy: (a) Reputation (b) Reputation update accuracy

We first compare the reputation decrease speed of the four mechanisms. The percentage of adversaries among all users is set from 0 to 50%. The results are shown in the Fig. 3 (a). It is clear that as the ratio of adversaries increases, the reputation in all the four mechanisms drops, while the reputation in the CCRM decreases the fastest. Because the adversaries are colluding and bad mouthing, the adversaries' false reports will gain more support from other collusive users and they can also provide dishonest recommendations to frame up good parties and/or

boost the reputation values of malicious peers. However, the VDCR and VCG-RIM schemes proposed in the CCRM and RP-CRM, respectively, can incentivize users to tell truth to effectively defend against collusion attacks and bad mouthing attacks, which makes the reputation decrease faster than under the ARTSense and Harmony mechanisms. Moreover, with the combination of security level-based data categories and user context information sensing, the CCRM implements fine-grained reputation evaluation that makes the reputation decrease faster than under the RP-CRM with adversaries in the network.

We also compare the reputation update accuracy of the four mechanisms. The percentage of adversaries is set at 30%. From the results shown in Fig. 3 (b), we can see that the users' reputation increases as time increases (i.e., the number of interactions increases). Similar to the speed of reputation decrease, the CCRM has a higher reputation update accuracy than the other three mechanisms. Although the RP-CRM also can incentivize users to provide truthful information to defend against internal collusion attacks and bad mouthing attacks, the absence of fine-grained reputation evaluation makes its reputation update accuracy worse than that of CCRM. Because both ARTSense and Harmony lack reliable recommendation evaluation mechanisms to effectively defend against internal collusion attacks and bad mouthing attacks, the false recommendations of malicious peers make the accuracy of updates of users' reputations lower than that under the RP-CRM and CCRM.

5.2.3 Impact of Context Information

Next, we compare the impact of context sensing on reputation among these four mechanisms. The characteristics of the mobile and random access in the MCC-based big data applications make contextual information such as time, location, data category, etc., an important factor in both the direct and the recommended reputation computation processes. Accurate contextual information can help improve the accuracy and reliability of the reputation evaluation. In this simulation, we consider two scenarios: (1) an honest network environment without attacks, and (2) a hostile network environment with adversaries.

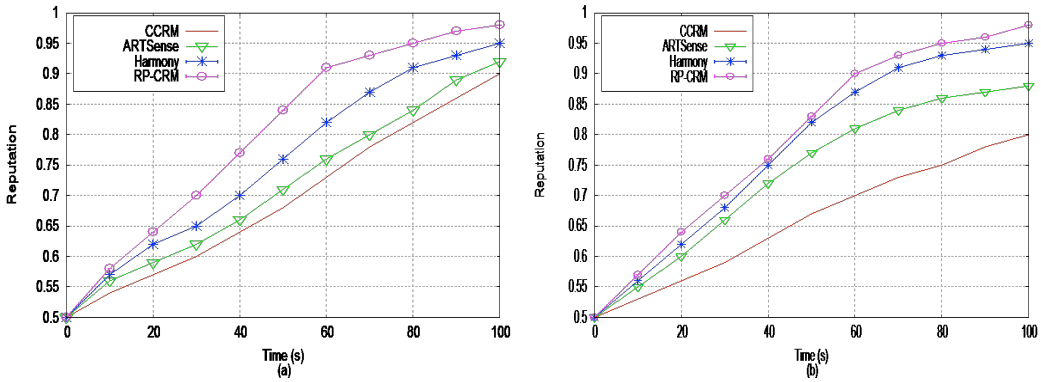

Fig. 4. Impact of Contextual Information: (a) Honest network (b) Hostile network

The results in Fig. 4 (a) show that users' reputations increase as the interaction time increases because they always provide honest information in the honest network. In the RP-CRM, the evaluation and update of reputations only depend on the historical interaction behaviors and ignore the time and location influences, so the reputation of the RP-CRM increases most quickly among the four mechanisms. For Harmony, although it introduces the data category into the reputation evaluation, time and location influence are ignored, which makes its reputation evaluation less accurate than that of CCRM and ARTSense. On the other hand, ARTSense considers the time and location influence on reputation, but it does not take the data category into account, so its reputation evaluation accuracy is lower than that of the CCRM. In Fig. 4 (b), we observe that although the adversaries exist, the value and increase speed of reputation of the RP-CRM and Harmony are close to those in Fig. 4 (a). This is because they cannot effectively discover and defend against context-based attacks. In contrast, owing to the introduction of contextual information into the reputation evaluation, the ARTSense and CCRM can overcome the shortcomings of the RP-CRM and Harmony; hence their value and increased speed of reputation are less than those in Fig. 4 (a). Meanwhile, the comprehensive consideration of the time, location and data category makes the performance of the CCRM better than the ARTSense.

5.2.4 Effective Recommendation Rate

We also evaluate the effectiveness and reliability of the four reputation mechanisms by comparing their ERR performances.

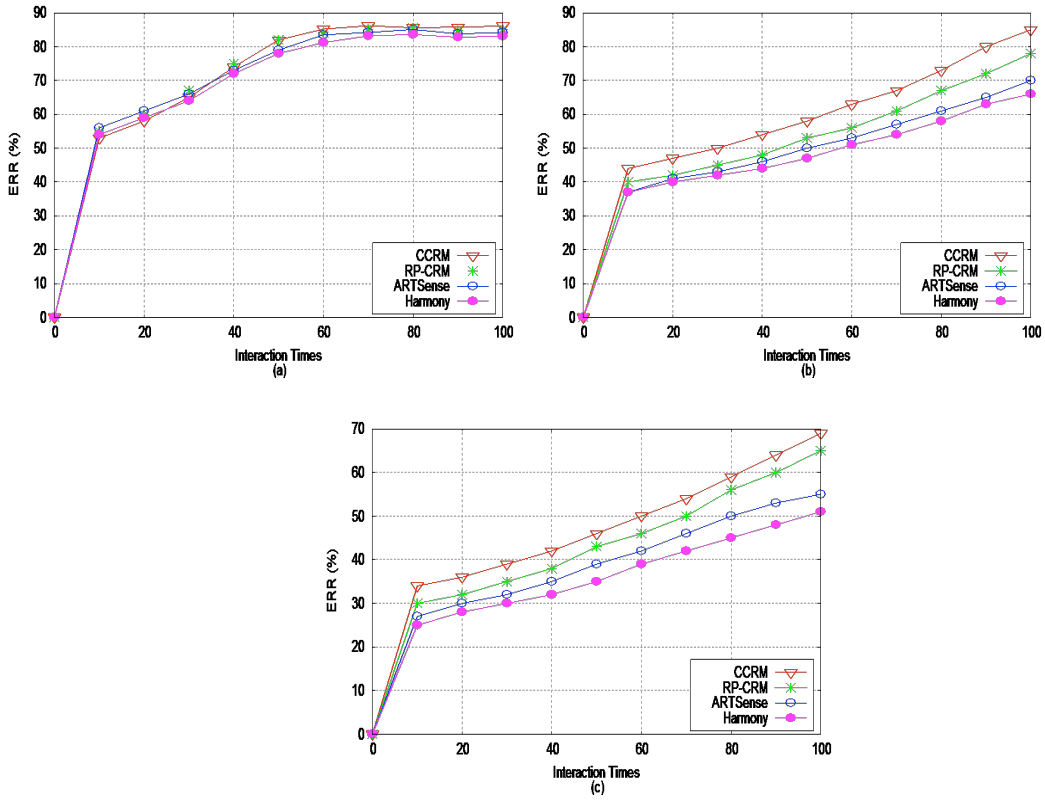

Fig. 5. Effective recommendation rate: (a) in an honest network, (b) in a hostile network with 20% adversaries, and (c) in a hostile network with 50% adversaries

The results in Fig. 5 (a) show that under the honest network environment, the ERRs of the four mechanisms are very close. Compared to the results in Fig. 5 (a), in Fig. 5 (b) and (c) where the adversaries are present, the ERRs of the ARTSense and Harmony decrease by 15% and 30%, respectively, and the ERRs of the RP-CRM and CCRM decrease by 10% and 20%, respectively. The comparison results show that the internal attacks have a large impact on the effectiveness and reliability of the recommendation, which will further affect the effectiveness and reliability of the final reputation evaluation. In Fig. 5 (b) and (c), the adversaries may launch the collusion attacks and bad mouthing attacks to provide the false recommendation information, which makes the ERR performance worse than that in the honest network.

In addition, Fig. 5 (b) and (c) show that the ERR of the CCRM is higher than the other three mechanisms. The reason is that the VDCR in CCRM can effectively incentivize the recommenders to tell the truth and thus enhance the reliability of recommendations. Moreover, the security relevance factor ensures that only those recommendations with similar security levels and related historical data access categories are accepted, which further improves the effectiveness of the recommendation. For RP-CRM, although it selects the recommendation

user and the recommendation based on the security relevance factor, it cannot incentivize users to tell the truth, which results in a lower ERR than the CCRM. ARTSense and Harmony did not consider improving the effectiveness and reliability of recommendations, so their ERRs are worse than those of the CCRM and RP-CRM.

5.2.5 Malicious User Detection Rate

Next, we analyze the malicious user detection rate under two hostile network environments with 50% and 70% adversaries, respectively.

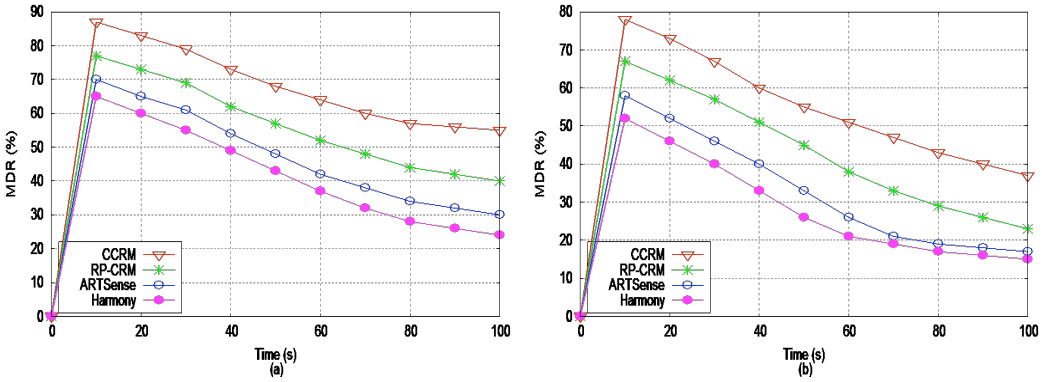

Fig. 6. Malicious user detection rate: (a) with 50% adversaries and (b) with 70% adversaries

In Fig. 6 (a) and (b), as expected, the MDR decreases with time and the growing percentage of adversaries. It is observed that the MDR of the CCRM is highest among the four mechanisms. This finding is observed because the integrated combination of the data category, context sensing, security relevance and recommendation incentive scheme improves the accuracy, efficiency, and reliability of the reputation evaluation and thus enhances the MDR. Although the other mechanisms also adopt related technologies to improve the accuracy and reliability of the reputation evaluation, they either consider the improvement of the direct reputation evaluation or the improvement of the recommended reputation evaluation in isolation. Therefore, their MDR is lower than that of the CCRM.

5.2.6 Sensitivity Analysis

In this subsection, we provide a sensitivity analysis of the parameters α_c, $\alpha_{location}$ and α_{time} that strongly characterize the proposed CCRM.

In Figs. 7-9, we can see that the user's reputation first increases and then starts to fluctuate as the parameters α_c, $\alpha_{location}$ and α_{time} increase. According to Eq. (2), the increase of the

three parameters leads to the decrease of uncertainty, which improves the direct reputation and thus results in the increase of the final reputation when the recommended reputation is fixed. Hence the node's reputation increases when these three parameters increase initially. According to Eq. (1), however, the increase of these three parameters may also lead to the decline of the direct reputation because 1) with the higher α_c, the cardinality of data categories set **C** increases; 2) with the larger $\alpha_{location}$, the expected distance between the locations of the interacting users become smaller, and hence, the number of unsuccessful data access attempts, AC_f^c, increases; and 3) with the bigger α_{time}, the expected time interval between the user interactions becomes larger, and hence, the number of unsuccessful data access attempts, AC_f^c, increases. Consequently, with the declining direct reputation, the final reputation becomes more dependent on and starts to fluctuate in line with the recommended reputation.

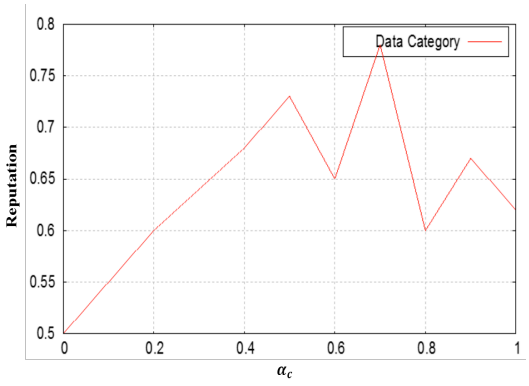

Fig. 7. Impact of data category access factor α_c on the reputation

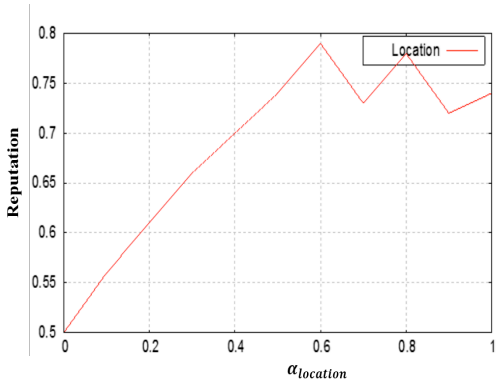

Fig. 8. Impact of location factor $\alpha_{location}$ on the reputation

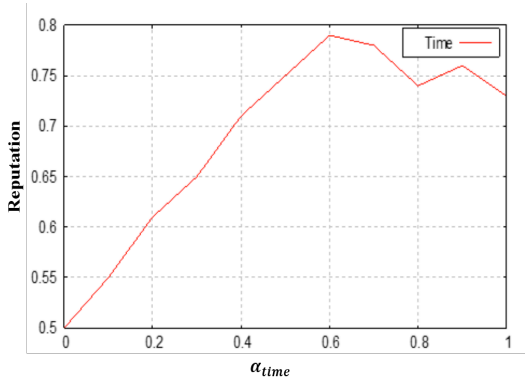

Fig. 9. Impact of time factor α_{time} on the reputation

5.3 Discussion of the Results

In this subsection, we discuss the simulation experimental results to offer some insights obtained from the proposed approach.

1) Utility of the recommender

The results show that the more truthful information the recommender provides, the larger average utility it gets. According to the characteristics of VCG, a user that participates in the recommendation and does truth-telling is the best strategy; thus, a normal user in the honest work will always provide truthful information. The results show that the average utility of the malicious user is affected by providing dishonest recommendations. When the adversary gives dishonest recommendations, according to the characteristics of the VCG mechanism, the proposed VDCR makes it pay more, which leads to a decline in its utility.

2) Reputation and its update accuracy

As the ratio of adversaries increases, the reputation drops, while the reputation in the CCRM decreases the fastest compared to other mechanisms. The VDCR scheme proposed in the CCRM can incentivize users to tell the truth to effectively defend against the collusion attacks and bad mouthing attacks, which causes the reputation to decrease rapidly. Moreover, with the combination of security level-based data categories and user contextual information sensing, the CCRM implements fine-grained reputation evaluation. We can also see that the users' reputations increase as the time increases. Similar to the decreasing speed of reputation, the CCRM has a higher reputation update accuracy than the other three mechanisms.

3) Impact of context information

Owing to the introduction of contextual information into the reputation evaluation, the CCRM can overcome the shortcomings of the RP-CRM and Harmony and can obtain a smaller value and increased speed of reputation than the case where no adversary is present. Meanwhile, the comprehensive consideration of the time, location and data category makes the performance of the CCRM better than that of ARTSense.

4) Effective recommendation rate (ERR)

The comparison results show that the internal attacks have a large impact on the

effectiveness and reliability of the recommendation, which will further affect the effectiveness and reliability of the final reputation evaluation. We can also see that the ERR of the CCRM is higher than the other three mechanisms. Moreover, the security relevance factor ensures that only those recommendations with similar security levels and related historical data access categories are accepted, which further improves the effectiveness of the recommendation.

5) Malicious user detection rate (MDR)

The MDR decreases with the time and the growing percentage of the adversaries. It is observed that the MDR of the CCRM is the highest among all the four mechanisms. This is because the integrated combination of the data category, context sensing, security relevance evaluation and recommendation incentive scheme improves the accuracy, efficiency, and reliability of the reputation evaluation, and thus enhances the MDR.

6) Parameters sensitivity

We can see that the user's reputation first increases and then starts to fluctuate as the parameters α_c, $\alpha_{location}$ and α_{time} increase. The increase of the three parameters leads to the decrease of the uncertainty, which improves the direct reputation and thus results in the increase of the final reputation. The further increase of these three parameters, however, may also lead to the decline of the direct reputation, which causes the final reputation to become more dependent on the recommended reputation and begin to fluctuate in line with the recommended reputation.

6. Conclusions

In this paper, we investigated the problem of protecting against internal attacks for enhancing data veracity in Mobile Cloud Computing (MCC). A new category-based context aware and recommendation incentive reputation mechanism named CCRM has been proposed, which incorporates innovative technologies in terms of data categories, context sensing, security relevance and recommendation incentive. The simulation-based experiments and performance analysis have verified that the CCRM is effective and efficient. More specifically, in the presence of collusion attacks and bad mouthing attacks, the utility of the recommender, the decrease speed and update accuracy of reputation, the effective recommendation rate, and the malicious user detection rate of the proposed CCRM are better

than those of the existing RP-CRM, ARTSense and Harmony mechanisms.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (61363068, 61472083, 61671360), the Pilot Project of Fujian Province (formal industry key project) (2016Y0031), the Foundation of Science and Technology on Information Assurance Laboratory (KJ-14-109) and the Fujian Provincial Key Laboratory of Network Security and Cryptology Research Fund.

References

[1] M. Ali, S. U. Khan, A. V. Vasilakos, Security in Cloud Computing: Opportunities and Challenges, Information Sciences, 305(2015) 357-383.

[2] S. Aman, C. Chelmis, V. Prasanna, Addressing data veracity in big data applications, The 2014 IEEE International Conference on Big Data (Big Data), (2014) 1-3.

[3] E. Bertino, Data security – challenges and research opportunities, in: Secure Data Management. Springer, (2014) 9-13.

[4] T. Bodnar, C. Tucker, K. Hopkinson, et al, Increasing the veracity of event detection on social media networks through user trust modeling, The 2014 IEEE International Conference on Big Data (Big Data), (2014) 636-643.

[5] S. Chen, G. Wang, W. Jia, κ -FuzzyTrust: Efficient Trust Computation for Large-Scale Mobile Social Networks Using a Fuzzy Implicit Social Graph, Information Sciences, 318(2015)123-143.

[6] A. Comi, L. Fotia, F. Messina, et al, A partnership-based approach to improve QoS on federated computing infrastructures, Information Sciences, 367–368 (2016) 246-258.

[7] J. Deng, R. Q. Zhang, L. Y. Song, Z. Han, B. L. Jiao, Truthful mechanisms for secure communication in wireless cooperative system, IEEE Transactions on Wireless Communications, 12(9) (2013) 4236 – 4245

[8] A. Hammam, S. Senbel, A trust management system for ad-hoc mobile clouds, The 8th International Conference on Computer Engineering & Systems (ICCES), (2013) 31-38.

[9] A. J. Goldsmith, S. G. Chua, Variable rate variable power MQAM for fading channels, IEEE Transactions on Communication, 45(10) (1997) 1218-1230.

[10] J. Kepner, V. Gadepally, P. Michaleas, N. Schear, M. Varia, A. Yerukhimovich, R. K. Cunningham, Computing on masked data: a high performance method for improving big data veracity, The 2014 High Performance Extreme Computing Conference (HPEC), (2014) 1 - 6

[11] A. N. Khan, M. L. M. Kiah, S. U. Khan, S. A. Madani, Towards secure mobile cloud computing: a survey, Future Generation Computer Systems, 29 (5) (2013) 1278-1299.

[12] M. Kim, O. P. Sang, Trust management on user behavioral patterns for a mobile cloud computing, Cluster Computing, 16 (4) (2013) 725–731.

[13] H. Lin, J. Hu, J. F. Ma, L. Xu, L. Yang, CRM: a new dynamic cross-layer reputation computation model in wireless networks, The Computer Journal, 58(4) (2015) 656-667.

[14] H. Lin, L. Xu, Y. Mu, W. Wu, A reliable recommendation and privacy-preserving based cross-layer reputation mechanism for mobile cloud computing, Future Generation Computer Systems, 52 (2014) 125-136 doi:10.1016/j.future.2014.10.032.

[15] Y. Liu, K. Q. Li, Y. W. Jin, Y. Zhang, W. Y. Qu, A novel reputation computation model based on subjective logic for mobile ad hoc networks, Future Generation Computer Systems, 27(5) (2011) 547-554.

[16] T. Lukoianova, V. Rubin, Veracity roadmap: is big data objective, truthful and credible?, Advances In Classification Research Online, 24(1) (2013). doi:10.7152/acro.v24i1.14671

[17] C. Martinez-Cruz, C. Porcel, J. Bernabé-Moreno, et al, A model to represent users trust in recommender systems using ontologies and fuzzy linguistic modeling[J]. Information Sciences, 2015, 311(C):102-118.

[18] N. Mohammed, H. Otrok, L. Wang, Mechanism design-based secure leader election model for intrusion detection in MANET, IEEE Transactions on Dependable and Secure Computing, 8(1) (2011) 89-103.

[19] J. Ren, Y. Zhang, K. Zhang, et al, Exploiting mobile crowdsourcing for pervasive cloud services: challenges and solutions, IEEE Communications Magazine, 53(3) (2015) 98-105.

[20] J. Sanger, C. Richthammer, S. Hassan, et al, Trust and big data: a roadmap for research, The 25th International Workshop on Database and Expert Systems Applications (DEXA), (2014) 278-282.

[21] A. S. Sethi, V. Y. Hnatyshin, The practical OPNET user guide for computer network simulation, CRC Press, 2012.

[22] H. Shen, G. Liu, An efficient and trustworthy resource sharing platform for collaborative cloud computing, IEEE Transactions on Parallel and Distributed Systems, 25(4) (2014) 862-875.

[23] X. Wang, W. Cheng, P. Mohapatra, et al, Artsense: anonymous reputation and trust in participatory sensing, The 2013 IEEE Proceedings of INFOCOM, (2013) 2517-2525.

[24] Z. Wei, W. Zhou, M. Kang, M. Collins, P. Nixon, A strategy-proof trust mechanism for pervasive computing environments, The 6th International Conference on Mobile Adhoc and Sensor Systems(MASS), (2009) 728-733.

[25] Z. Yan, W.X. Ding, X.X. Yu, H.Q. Zhu, R. H. Deng, Deduplication on Encrypted Big Data in Cloud, IEEE Transactions on Big Data, 2(2) 2016 138-150.

[26] Z. Yan, X.Y. Li, R. Kantola, Controlling Cloud Data Access Based on Reputation, Mobile Networks and Applications, 20 (6) (2015) 828-839.

[27] Z. Yan, X.Y. Li, M.J. Wang, A.V. Vasilakos, Flexible Data Access Control based on Trust and Reputation in Cloud Computing, IEEE Transactions on Cloud Computing, 2015. Doi: 10.1109/TCC.2015.2469662.

[28] Z. Yan, M.J. Wang, Y.X. Li, A.V. Vasilakos, Encrypted Data Management with Deduplication in Cloud Computing, IEEE Cloud Computing Magazine, 3(2) (2016) 28-35.

[29] S. Yin, O. Kaynak, Big data for modern industry: challenges and trends, Proceedings of the IEEE, 103(2) (2015) 143-146.

[30] Y. Zhang, M. Schaar, Incentive provision and job allocation in social cloud systems, IEEE Journal on Selected Areas in Communications, 31(9) (2013) 607-617.

[31] C. Zhu, V. C. M. Leung, L. T. Yang, L. Shu, Collaborative Location-based Sleep Scheduling for Wireless Sensor Networks Integrated with Mobile Cloud Computing, IEEE Transactions on Computers, 64(7) (2015) 1844 -1856.

[32] C. Zhu, Z. Sheng, V. C. M. Leung, L. Shu, L. T. Yang, Towards Offering More Useful Data Reliably to Mobile Cloud from Wireless Sensor Network, IEEE Transactions on Emerging Topics in Computing , 3 (1) (2015) 84-94.

[33] C. Zhu, H. Nicanfar, V. C. M. Leung, L. T. Yang, An Authenticated Trust and Reputation Calculation and Management System for Cloud and Sensor Networks Integration, IEEE Transactions on Information Forensics and Security, 10 (1) (2015) 118-131.

[34] D. Zissis, D. Lekkas, Addressing cloud computing security issues, Future Generation Computer Systems, 28 (3) (2012) 583–592.