LETTER

Efficient Privacy-Preserving Reputation Evaluation in Decentralized Environments*

Youwen ZHU[†a)], ***Nonmember and*** **Tsuyoshi TAKAGI**[†], ***Member***

SUMMARY A decentralized secure protocol for casting trust rating in reputation systems (StR protocol) is lately proposed by Dimitriou and Michalas, and the StR protocol is verified to be faster than the previous work providing anonymous feedback. In this letter, we present new enhanced scheme of StR. Compared with StR protocol, our new approach attains the exactly same security, but requires less processing time and about half communication overheads. Therefore, we improve the performance without sacrificing any security, especially the communication delay is dramatically reduced.

key words: *reputation evaluation, decentralized environment, privacy-preserving, communication delay*

1. Introduction

With the prosperous development of online communities, reputation systems [1] have been widely adopted to assist users to avoid interacting with untrustworthy or unreliable nodes. In reputation systems, the behavior of an entity is evaluated according to the quality of service he provided. The reputation systems utilize the feedback information as the evidence, then aggregate other users' feedback to decide whether one entity is reliable.

Nevertheless, the users may not provide genuine feedback due to some realistic reasons, such as worrying about possible retaliation [2]. For the problem, an obvious solution is that the feedbacks are provided in anonymous or privacy-preserving manner such that nobody apart from the provider can learn the recommended information. In recent years, several works [3]–[6] have addressed the privacy concerns of feedback providers in decentralized environments where no trusted authority exists. The paper [3] proposed three protocols to protect privacy of feedback in decentralized additive reputation system, but the schemes can resist the collusion attack of $(n-1)$ users with only a certain probability at best. Here, n is the number of feedback providers. Hasan et al. [4] presented another privacy-preserving protocol in which each user splits his private vote into k shares $(k < n-1)$, selects k trustworthy agents, and sends one share to each selected agent. The accumulated protocol (AP) and weighted accumulated protocol (WAP) are put forward by [5] to compute trust in a completely distributed manner. The schemes in [5] only require the transmission of $O(n)$ messages, but the length of one message is as big as $O(n)$. Besides, the computation cost of them is inefficiently high. Dimitriou and Michalas [6] lately proposed an efficient multiparty trust computation approach (StR) in decentralized environments such as ad hoc networks. StR is proven to be secure against the collusion of querying node and at most $k < (n-1)$ users out of the feedback providers. Their implementation results show the communication delay of StR is about 11% of that of the protocol in [5]. Through using the zero-knowledge proofs [7], the work [6] further gives the extension version StR^M with security against malicious adversaries. StR^M is about 7 times slower than its original version under semi-honest adversaries model, since StR^M requires about sevenfold messages transmission.

In this letter, we analyze the efficient lately proposed scheme – StR [6], and observe that a higher efficiency can be achieved without sacrificing any security. Using the random splitting approach, we present a new secure reputation approach in decentralized environment. Our scheme can attain the same security with StR, that is, it can resist the collusion attack of querying node and at most $k < (n-1)$ feedback providers. Compared to StR, our new approach requires less computation cost, and decreases communication overheads by 50% or so. We achieve the performance improvements through analyzing the splitting approach and eliminating the computation and communication redundancy in StR. Our proposed scheme only considers the semi-honest adversaries model, but it can be easily converted into an equivalent protocol resisting malicious adversaries by making use of the zero-knowledge proofs [7]. The converting manner is similar to that for extending StR into StR^M, thus, this letter restricts our attention to only the semi-honest adversaries model.

The rest of the letter is organized as follows. Section 2 describes the problem and adversaries model, then reviews and discusses the StR protocol. In Sect. 3, we propose our new secure reputation scheme, followed by analysis and comparison with StR. At last, Sect. 4 concludes the letter.

2. Problem Statement and Our Discussion of Dimitriou et al.'s StR Protocol

2.1 Problem Statement and Adversaries Model

Problem Statement: Concretely speaking, this letter deals with the following situation: A querying user A_q has no

Manuscript received June 18, 2013.

[†]The authors are with the Institute of Mathematics for Industry, Kyushu University, Fukuoka-shi, 819-0395 Japan.

*This work was supported in part by the Japan Society for the Promotion of Science fellowship (No. P12045), the National Natural Science Foundation of China (No. 61272404).

a) E-mail: ywzhu@imi.kyushu-u.ac.jp

DOI: 10.1587/transfun.E97.A.409

enough information about a target node A_t. Then, A_q selects the node set $\boldsymbol{U} = \{U_1, U_2, \cdots, U_n\}$, and asks each node U_i to provide his feedback (v_i) about A_t. The challenging problem is that A_q would like to obtain the aggregated feedback sum ($\sum_{i=1}^{n} v_i$) while no node wants his feedback to be learned by others. That is, v_i should be kept privately to U_i throughout the reputation evaluation. The feedback v_i in this problem is also called the vote of U_i. In this letter, we use feedback and vote interchangeably.

Adversaries Model: During the reputation evaluation, the participants (querying user and feedback providers) in our scheme are assumed to be semi-honest, a.k.a., honest-but-curious. Under the semi-honest adversaries model, each node will strictly follow the protocol steps, but keep a record of all data he learns during the protocol to infer as much information about the other parties as possible. Beyond the semi-honest behaviors, several participants may collude, and the colluding nodes will share all information they learn to infer as much information about other legitimate users' private feedbacks as possible.

We also assume the communication channel between any two participants are secure, which can be realized by conventional cryptography.

2.2 Review and Discussion of StR Protocol

In [6], Dimitriou and Michalas presented StR protocol to support decentralized reputation computation in the situation as our problem statement describes. After A_q selects the set $\boldsymbol{U}$ and distributes it to each U_i, StR consists of two rounds. First round aims at perturbing the private feedbacks, in which each U_i generates a random number r_i, splits it into n random shares: $r_i = r_{i1} + r_{i2} + \cdots + r_{in}$, and sends r_{ij} to U_j ($j \neq i$). Then, U_i sets the perturbed feedback as $b_i = v_i + r_i - (\sum_{j=1}^{n} r_{ij})$. In second round, U_i submits b_i to A_q, and A_q can obtain the final aggregate feedback sum $\sum_{i=1}^{n} v_i = \sum_{i=1}^{n} b_i$. It has been shown that if A_q is uncompromised, the collusion of all the other $(n-1)$ nodes ($\boldsymbol{U} \setminus \{U_i\}$) cannot infer any information about the private feedback v_i of U_i, since each U_j ($j \neq i$) only receives a random share of r_i but nothing about v_i. Furthermore, while the querying node A_q is compromised, StR can resist the collusion of A_q and at most $k < (n-1)$ feedback providers. Based on simulation experiments, [6] shows that StR is much more efficient than the previous scheme in [5].

However, we observe many messages in StR offer no benefit to privacy-preservation. Through eliminating the redundant messages and the corresponding computation overheads, we can attain reduction in communication and computation cost, especially the communication delay can be decreased by about 50%. Next, we will further discuss StR.

Definition 1: (*collusion-resistant degree*) In a secure reputation evaluation scheme $\boldsymbol{f}$, for the private feedback v_i of node U_i, if the collusion of any t nodes (out of querying node A_q and feedback providers apart from U_i) cannot infer v_i, but there exists a collusion of $(t+1)$ nodes which can infer the private v_i, then, we define t is the *collusion-resistant degree* of v_i in $\boldsymbol{f}$, and use $\boldsymbol{d}_{cr}(v_i)$ to denote it, i.e., $\boldsymbol{d}_{cr}(v_i) = t$.

Fig. 1 Messages transmission between nodes.

According to the above definition 1, we can obtain that the private feedback v_i is secure against the collusion of at most $\boldsymbol{d}_{cr}(v_i)$ nodes in the scheme $\boldsymbol{f}$. The bigger $\boldsymbol{d}_{cr}(v_i)$, the larger collusion v_i can resist. In our problem statement, there are in total $(n+1)$ participants: one querying user and n feedback providers. Thus, the maximum value of $\boldsymbol{d}_{cr}(v_i)$ is not bigger than n.

In StR [6], the collusion of ($\boldsymbol{U} \setminus \{U_i\}$) can infer nothing about v_i, and the collusion of A_q and $k < (n-1)$ feedback providers also cannot get v_i. That is, the collusion of any $(n-1)$ nodes cannot infer v_i. Nevertheless, A_q can figure out $v_i = (\sum_{j=1}^{n} b_j) - (\sum_{j=1}^{i-1} v_j + \sum_{j=i+1}^{n} v_j)$ through colluding with all the $(n-1)$ nodes in ($\boldsymbol{U} \setminus \{U_i\}$). Therefore, $\boldsymbol{d}_{cr}(v_i) = n-1$.

Definition 2: (*utility per message*) In a secure reputation evaluation scheme $\boldsymbol{f}$, if m is total number of transmission messages of all the n feedback providers. Then, we define the *utility per message* ($\boldsymbol{\mu}$) as $\boldsymbol{\mu} = \frac{1}{m} \sum_{i=1}^{n} \boldsymbol{d}_{cr}(v_i)$.

In the StR protocol [6], each U_i sends $(n-1)$ messages in first round and one message in second round, respectively. Then, the total number of messages is $n * (n-1+1) = n^2$. Hence, the *utility per message* of StR can be obtained $\boldsymbol{\mu} = \frac{1}{n^2} \sum_{i=1}^{n} \boldsymbol{d}_{cr}(v_i) = \frac{n*(n-1)}{n^2} = \frac{n-1}{n}$.

We further discuss the utility of each message. As shown in Fig. 1(a), if there is one message transmission between U_i and U_j (without loss of generality, we assume U_i sends his random share r_{ij} to U_j), then, U_i and U_j can use r_{ij} to perturb their feedbacks, and any collusion that does not include U_j (resp. U_i) cannot infer v_i (resp. v_j) because of the randomness of r_{ij}. Therefore, one message (r_{ij}) transmission can improve *collusion-resistant degree* of both v_i and v_j, compared to no message transmission between them. However, after r_{ij}, the second random share r_{ji} transmission can not enhance the *collusion-resistant degree* of any one, since the adversaries aiming at figuring out v_i (resp. v_j) can simultaneously obtain r_{ij} and r_{ji} through corrupting U_j (resp. U_i). Therefore, half of messages in first round of StR offer no benefit to the security. Through eliminating the useless messages transmission, Sect. 3 presents our new scheme which is as secure as StR, but achieves higher implementation efficiency than StR.

3. Our Scheme

This section presents a basic protocol: Secure Reputation Evaluation Protocol (SREP) and the enhanced version: Communication-balanced Secure Reputation Evaluation Protocol (CbSREP), both of which hold the same security with StR. Compared to StR, SREP nearly decreases

communication overheads by 50%; through balancing communication load of nodes, CbSREP achieves about 50% reduction in communication delay.

3.1 Basic Protocol

In the first scheme, we avert the useless messages transmission by the rule: during perturbing the feedback (first round), U_i $(1 < i \leqslant n)$ only sends his random shares to U_j $(j < i)$, and U_1 sends nothing and only waits to receive the random shares destined to him. Then, each node uses the random numbers he sends and receives to perturb his sensitive feedback. In second round, U_i directly uploads his blinded feedback to A_q, and the querying user can obtain the aggregate reputation by summing n perturbed votes. The concrete steps are described in protocol 1.

Protocol 1 Secure Reputation Evaluation Protocol (SREP)

1: A_q selects and distributes the set $\boldsymbol{U} = \{U_1, U_2, \cdots, U_n\}$.
2: **Round 1 – all the n feedback providers in parallel**:
3: **for all** $U_i \in \boldsymbol{U}$ **do**
4: U_i generates $(i-1)$ random numbers $\{r_{i1}, r_{i2}, \cdots, r_{i,i-1}\}$.
5: **for all** $j < i$ **do**
6: U_i sends r_{ij} to U_j.
7: **end for**
8: U_i waits until he receives the random numbers from all U_k $(i < k \leqslant n)$, and computes the perturbed feedback $b_i = v_i + (\sum_{j=1}^{i-1} r_{ij}) - (\sum_{k=i+1}^{n} r_{ki})$.
9: **end for**
10: **Round 2 – all the n feedback providers in parallel**:
11: **for all** $U_i \in \boldsymbol{U}$ **do**
12: U_i sends b_i to A_q.
13: **end for**
14: At last, A_q computes the aggregate reputation $\sum_{i=1}^{n} b_i$.

In protocol 1, for each r_{ij}, U_i adds it to b_i, and U_j subtracts it from b_j. Hence, after first round, we have the equation $\sum_{i=1}^{n} b_i = \sum_{i=1}^{n} v_i$ which ensures the correctness of A_q's aggregate result. For the security of our basic scheme, we have the following theorems.

Theorem 1: (Uncompromised A_q) In protocol 1, for any $i \in \{1, 2, \cdots, n\}$, the collusion of $(\boldsymbol{U} \setminus \{U_i\})$ cannot deduce anything about v_i.

Proof: While A_q is uncompromised, for the information about data of U_i, the nodes of $(\boldsymbol{U} \setminus \{U_i\})$ can get nothing but the $(i-1)$ random numbers $\{r_{i1}, r_{i2}, \cdots, r_{i,i-1}\}$ throughout the protocol 1. Therefore, the collusion of $(\boldsymbol{U} \setminus \{U_i\})$ cannot deduce anything about the private vote v_i of U_i. □

Theorem 2: (Compromised A_q) In protocol 1, for any $i \in \{1, 2, \cdots, n\}$, the collusion of A_q and $k < (n-1)$ nodes out of $(\boldsymbol{U} \setminus \{U_i\})$ cannot infer v_i.

Proof: If A_q is also compromised, we consider the collusion attack of A_q and any $k < (n-1)$ nodes out of $(\boldsymbol{U} \setminus \{U_i\})$ as follows.

Without loss of generality, we assume the k colluding feedback providers does not include $U_j \in (\boldsymbol{U} \setminus \{U_i\})$. Obviously, $j \neq i$. If $i > j$, U_i has sent r_{ij} to U_j in the round 1; if $i < j$, U_i can receive the random r_{ji} from U_j. Then, U_i has used r_{ij} (or r_{ji}) to perturb v_i. While A_q and $k < (n-1)$ feedback providers collude, the adversaries can at most figure out $(v_i + r_{ij})$ if $i > j$, or $(v_i - r_{ji})$ if $i < j$. Nevertheless, only U_i and U_j learn the value of random number r_{ij} or r_{ji}, hence, the adversaries cannot infer v_i from $(v_i + r_{ij})$ or $(v_i - r_{ji})$. It completes the proof of theorem 2. □

Theorems 1 and 2 guarantee that protocol 1 is as secure as StR in [6]. For any $i \in \{1, 2, \cdots, n\}$, *collusion-resistant degree* of v_i in protocol 1 is also $(n-1)$, i.e., $\boldsymbol{d}_{cr}(v_i) = n-1$, and U_i sends $(i-1+1)$ messages in total. Then, the number of all messages in protocol is $\frac{n(n+1)}{2}$ and the *utility per message* of protocol 1 is $\boldsymbol{\mu} = \frac{2}{n(n+1)} \sum_{i=1}^{n} (n-1) = \frac{2(n-1)}{n+1}$. Therefore, $(\frac{n(n+1)}{2})/n^2 \approx 50\%$ and $(\frac{2(n-1)}{n+1})/(\frac{n-1}{n}) \approx 2$. Protocol 1 can decrease the communication overheads by 50% and nearly double the *utility per message*. Additionally, the nodes in our protocol totally generate $\frac{n(n-1)}{2} = \sum_{i=1}^{n} (i-1)$ random numbers, which is about one half of n^2 in StR, and protocol 1 requires less computation cost.

3.2 Communication-balanced Protocol

As we have shown, our protocol 1 can decrease computation cost, and achieve about 50% reduction in communication overheads. While all feedback providers implement the scheme in parallel, the total time cost mainly depends on the most time-consuming node. In the above protocol, U_n generates $(n-1)$ random numbers and sends $(n-1)$ messages in first round, which is almost the same as StR. Then, though protocol 1 reduces the overheads of other nodes, it cannot significantly mitigate the communication delay. In this sub-section, we further balance the messages transmission of the nodes, and present a new enhanced scheme to effectively reduce communication delay.

Let $\lceil \frac{n-1}{2} \rceil$ be the smallest integer which is not less than $\frac{n-1}{2}$. In our enhanced protocol, each U_i selects $\lceil \frac{n-1}{2} \rceil$ random numbers and distributes them to the $\lceil \frac{n-1}{2} \rceil$ clockwise successor nodes shown in Fig. 2. For each random number, the sender adds it to his blinded vote and the receiver subtracts it from his perturbed feedback, based on which the sum of perturbed votes will just equal to that of original ones. After completing the perturbation, A_q will collect and

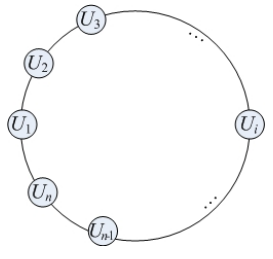

Fig. 2 Feedback providers ring.

sum the blinded feedbacks in second round. The detail steps are shown in protocol 2.

Protocol 2 Communication-balanced Secure Reputation Evaluation Protocol (CbSREP)

1: A_q selects and distributes the set $\boldsymbol{U} = \{U_1, U_2, \cdots, U_n\}$.
2: **Round 1 – all the n feedback providers in parallel**:
3: **for all** $U_i \in \boldsymbol{U}$ **do**
4: U_i sets the initial value $b_i = v_i$.
5: **for all** $j = 1, 2, \cdots, \lceil \frac{n-1}{2} \rceil$ **do**
6: $x = i + j$;
7: **if** $x > n$ **then**
8: $x = x - n$;
9: **end if**
10: U_i selects a random number r_{ix}, and sends r_{ix} to U_x.
11: U_i computes $b_i = b_i + r_{ix}$.
12: **end for**
13: U_i waits and computes the sum of $\lceil \frac{n-1}{2} \rceil$ random numbers destined to him. Let a_i equal to the sum. Then, U_i obtains the perturbed vote $b_i = b_i - a_i$.
14: **end for**
15: **Round 2 – all the n feedback providers in parallel**:
16: **for all** $U_i \in \boldsymbol{U}$ **do**
17: U_i sends b_i to A_q.
18: **end for**
19: At last, A_q computes the aggregate reputation $\sum_{i=1}^{n} b_i$.

We analyze the security of protocol 2 as follows.

Theorem 3: In protocol 2, for any $1 \leqslant i, j \leqslant n, i \neq j$, at least one random number is transported between U_i and U_j.

Proof: Without loss of generality, we assume $i < j$.
If $j \leqslant i + \lceil \frac{n-1}{2} \rceil$, then, U_i will send r_{ij} to U_j.
If $j > i + \lceil \frac{n-1}{2} \rceil$, then, $j > 1 + \lceil \frac{n-1}{2} \rceil$ and we have,

$$j + \left\lceil \frac{n-1}{2} \right\rceil > 1 + 2 * \left\lceil \frac{n-1}{2} \right\rceil \geqslant 1 + (n-1) = n.$$

Hence, U_j will send one random number to each node in the set $\{U_{j+1}, U_{j+2}, \cdots, U_n, U_1, U_2, \cdots, U_{j+\lceil \frac{n-1}{2} \rceil - n}\}$. Since $j + \lceil \frac{n-1}{2} \rceil - n > i + 2 * \lceil \frac{n-1}{2} \rceil - n \geqslant i + (n-1) - n = i - 1$, there is $j + \lceil \frac{n-1}{2} \rceil - n \geqslant i$. Hence, U_j will send r_{ji} to U_i.

In general, at least one random number is transported between any pair of nodes in $\boldsymbol{U}$. □

Theorem 4: (Uncompromised A_q) In protocol 2, for any $i \in \{1, 2, \cdots, n\}$, the collusion of $(\boldsymbol{U} \setminus \{U_i\})$ cannot infer anything about v_i.

Proof: In the protocol 2, the collusion of $(\boldsymbol{U} \setminus \{U_i\})$ can get nothing but $\lceil \frac{n-1}{2} \rceil$ random numbers about the information of U_i. Thus, they cannot infer anything about v_i. □

Theorem 5: (Compromised A_q) In protocol 2, for any $i \in \{1, 2, \cdots, n\}$, the collusion of A_q and $k < (n-1)$ nodes out of $(\boldsymbol{U} \setminus \{U_i\})$ cannot deduce v_i.

Proof: If compromised A_q collude with $k < (n-1)$ nodes out of $(\boldsymbol{U} \setminus \{U_i\})$, then, there exists at least one node $U_j \in (\boldsymbol{U} \setminus \{U_i\})$ and U_j is uncompromised. Theorem 3 shows one or more random numbers are transported between U_i and U_j. We assume X_{ij} is a random number that U_i (or U_j) sends to U_j (or U_i). Then, the collusion of A_q and k nodes at most can figure out $(v_i + X_{ij})$ (or $(v_i - X_{ij})$). Since the communication channel is secure between participants, only U_i and U_j can learn the random X_{ij}. Therefore, the aforementioned collusion cannot infer the value of v_i from $(v_i + X_{ij})$ (or $(v_i - X_{ij})$). At last, we obtain that theorem 5 holds. □

Similar to our basic protocol, protocol 2 also achieve the same security with StR owing to theorems 4 and 5. Nevertheless, each feedback provider in protocol 2 only sends $\lceil \frac{n-1}{2} \rceil$ messages in round 1 and one message in round 2. Thus, in our CbSREP, the communication overheads of each U_i are the same with that of any other node in $\boldsymbol{U}$, and they are $(\lceil \frac{n-1}{2} \rceil + 1)/n \approx 50\%$ of one node's communication cost in StR. Consequently, the communication delay of our protocol 2 will be about one half of that of StR. For the computation cost, each U_i in the new protocol only generates $\lceil \frac{n-1}{2} \rceil$ random numbers, thus protocol 2 takes less computation time than StR.

4. Conclusion

In this letter, we addressed the secure reputation evaluation in decentralized environment, and presented new high efficient scheme to preserve the privacy of each feedback. Analysis results show our scheme is as secure as the lately proposed StR protocol in [6], but our new approach decreases computation cost, and more importantly, our protocol can achieve about 50% reduction in communication delay.

References

[1] A. Jøsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," Decision Support Systems, vol.43, no.2, pp.618–644, 2007.
[2] P. Resnick and R. Zeckhauser, "Trust among strangers in internet transactions: Empirical analysis of ebay's reputation system," Advances in Applied Microeconomics, vol.11, pp.127–157, 2002.
[3] E. Pavlov, J.S. Rosenschein, and Z. Topol, "Supporting privacy in decentralized additive reputation systems," 2nd International Conference on Trust Management, LNCS 2995, pp.108–119, 2004.
[4] O. Hasan, L. Brunie, and E. Bertino, "k-shares: A privacy preserving reputation protocol for decentralized environments," 25th IFIP International Information Security Conference (SEC), pp.253–264, 2010.
[5] S. Dolev, N. Gilboa, and M. Kopeetsky, "Computing multi-party trust privately: In o (n) time units sending one (possibly large) message at a time," Proc. 2010 ACM Symposium on Applied Computing, pp.1460–1465, 2010.
[6] T. Dimitriou and A. Michalas, "Multi-party trust computation in decentralized environments in the presence of malicious adversaries," Ad Hoc Networks, 2013.
[7] O. Baudron, P.-A. Fouque, D. Pointcheval, J. Stern, and G. Poupard, "Practical multi-candidate election system," Proc. 20th Annual ACM Symposium on Principles of Distributed Computing, pp.274–283, 2001.