Methods for computing trust and reputation while preserving privacy

Ehud Gudes, Nurit Gal-Oz, and Alon Grubshtein

Dept. of Computer Science and Deutsche Telekom Labs at Ben-Gurion University, Beer-Sheva, 84105, Israel
ehud@cs.bgu.ac.il, galoz@cs.bgu.ac.il, alongrub@cs.bgu.ac.il

Abstract. Trust and Reputation systems in distributed environments attain widespread interest as online communities are becoming an inherent part of the daily routine of Internet users. Trust-based models enable safer operation within communities to which information exchange and peer to peer interaction are centric. Several models for trust based reputation have been suggested recently, among them the Knots model [5]. In these models, the subjective reputation of a member is computed using information provided by a set of members trusted by the latter. The present paper discusses the computation of reputation in such models, while preserving members' private information. Three different schemes for the private computation of reputation are presented, and the advantages and disadvantages in terms of privacy and communication overhead are analyzed.

1 Introduction

Recent years have seen a substantial growth of virtual communities across the Internet. The accessibility of information and services offered by these communities, makes it both possible and legitimate to communicate with strangers and carry out interactions anonymously, as rarely done in "real" life. However, virtual communities are prone to many types of deception ranging from people forging their identity and imposing as others, to people rating their service providers extremely high or extremely low unrelated to the service they have received. Trust and Reputation systems (TRS) provide communities with means to reduce the potential risk when communicating with people hiding behind virtual identities. These utilize the experience and knowledge accumulated and shared by all participants for assigning reputation values to individuals, and attempt to identify dishonest members and prevent their negative effect.

Reputation may depend on factors such as: interaction of multiple attributes (as in eBay), certainty of ratings [7], time of interaction and rating, and trust between members. The latter is crucial in obtaining reputation which is specifically compatible with a user profile or preferences. The issue of trust between members is discussed in [3, 5] among others. When anonymity is required, trust between members may be computed based on the similarity of their past ratings (see [5]). As more systems implement a trust mechanism between members the

impact of this information grows, and greater efforts should be made in keeping it private.

An empirical study conducted by [12] on data sets from eBay's reputation system reported a high correlation between buyer/seller ratings. Moreover, most feedback provided was positive. One explanation for these results is that when feedback providers' identities (pseudo-identities) are known, ratings are provided based on reasons of reciprocation and retaliation, not properly reflecting the trustworthiness of the rated parties. Thus preserving the trust level of members in each other private, while computing reputation becomes an important issue.

Decentralized reputation systems do not make use of a central repository to process reputation ratings [15] and are considered safer. In these, both the reputation of users and the ratings they give may be stored locally and known only to the corresponding user. The challenge here is to compute the reputation while maintaining private data. Its important to emphasize that based on the TRS, there exist different data which may be considered private, such as ratings, weights assigned to specific ratings, identity of the raters, trust between members, etc. A recent paper [9] suggested several privacy preserving schemes for simple computation of reputation (e.g. eBay [1]).

The current paper advances the state of the art as it uses an enhanced model of reputation computation which considers the trust members have in one another. We base our discussion on the *Knots* model [5] but the ideas presented may apply to any model in which trust between members is important in computing reputation. *Three* different methods for computing trust and reputation privately are presented, each offering a slightly different degree of privacy and communication overhead. All are analyzed with respect to users' malicious and non-malicious behavior, and the advantages of each are discussed. The rest of the paper is organized as follows. Section 2 provides an overview of related work and presents the knots model. Section 3 describes the three schemes and in Section 4 we conclude and give some future research directions.

2 Background and Related Work

The concern for privacy in communities in general, and the privacy of reputation information in particular, was discussed in several papers [8, 6, 14, 11, 4]. In [8] the authors discuss issues of privacy in a P2P network where reputation information is distributed among the P2P nodes. Requirements for fair use practices and their impact on system design, classes of information which may be leaked and managing the risks related to social and technical issues are all analyzed. However, no specific method for computing reputation is presented. In [6] a distributed trust management system, constructed from a two level hierarchy is described. The high level is composed of brokers which are responsible for aggregating trust information from their individual local nodes. This provides some privacy from one broker to another, although no privacy is provided at a single broker's network. Steinbrecher in [14] presents an information theoretic model of reputation privacy. She tries to model the amount of privacy lost when a single

user uses different pseudonyms in the same community or in different ones, or when a user changes her pseudonym. Her measure enables the estimate of *unlinkability* provided by such a pseudonym change. In [11], the authors discuss the issue of privacy for a user in multiple communities that requires a transfer of the reputation between these communities, creating the concept of cross-community reputation (CCR). CCR requirements were analyzed in [4] including issues of privacy, user control vs. community control, ontology matching and others.

Pavlov et al [9] provide the closest paper to the present work. It deals with private computation of reputation information, when the reputation is defined as an additive reputation system (e.g. the Beta reputation system [7]). The authors present three algorithms for computing additive reputation, with various degrees of privacy and with different level of protection against malicious users. A method for "witness selection" which reduces the risk of selecting dishonest witnesses is first presented. Then, a simple scheme which is very efficient but vulnerable to collusion of even two witnesses is introduced. The second scheme is more resilient towards curious users, but vulnerable to collusions and uses a secret splitting scheme. The last, provides the most secure protocol which uses the verifiable secret sharing scheme [10] based on Shamir's secret sharing scheme [13]. The complexity of the scheme is quite high and requires $O(n^3)$ messages where n is the number of contributing nodes. These schemes can be used with additive computations. Our work, which is described in terms of the Knots model [5] uses additional factor, which is not additive.

2.1 The Knots Model

The Knots model [5] is a TRS model for large-scale virtual communities. It is composed of three modules: *member trust inference module*, *Knots construction module*, and *reputation computation module*. The member trust inference module identifies trust relations among members; the Knots construction module utilizes these relations to generate trust Knots; the reputation computation module computes local reputations within Knots and global reputations for the whole community. Without loss of generality we adopt the notation from this model, but the underlying approach exists in other models as well.

- *TrustMember* (TM) is trust in the context of recommendations. It is a trust value that quantifies the extent by which one member relies on another member to rate experts "correctly".
- *TrustExpert* (TE) is trust in the context of experts. Specifically, it quantifies the extent by which a member relies on an expert to successfully provide the service it requires.
- An α - *Trust Set* of a member A, is the set of all members whom A trusts with level α or more.

The motivation for our current paper is to provide trust based reputation models such as the knot model, with means to compute reputation in a distributed manner while preserving private information. In this context the trust one member has in another (TM) and the trust a member has in an expert (TE), are

both considered private information. One would prefer not to reveal her trust in a member from which she gets recommendations. On the other hand the recommending party may prefer to keep her recommendation private due to the fear of retaliation. Revealing these to malicious parties exposes the community to the risk of manipulating recommendations.

3 Privacy preserving Trust and Reputation computation

The problem of privacy we address is the following: assume the existence of an expert x, and a member A. A needs to compute her trust in the expert, based on the experience other members of the community have had with this expert. We use the Trust-set as the group of members participating in this computation. The trust of A in x using Trust-sets [5] can be computed according to:

$$TE(A,x) = \frac{\sum_{B_i \in TrustSet(A),\ DTE(B_i,x)\neq\perp} DTE(B_i,x)\cdot TM(A,B_i)}{\sum_{B_i \in TrustSet(A),\ DTE(B_i,x)\neq\perp} TM(A,B_i)}$$

where:

- $DTE(B_i,x)$ - the trust member B_i has in expert x based on her own accumulated experience.
- $ITE(A,B_i,x)=TM(A,B_i)\cdot DTE(B_i,x)$ - the indirect trust member A has in expert x based on B_i's direct trust in x.

We assume that $TM(A,B)$ is known to agent A since it reflects her private information. Therefore the denominator is easy to compute by A without disclosing private information. The nominator is a sum of products of two terms, the first one is assumed to be distributed among the agents B_i, and known only by its corresponding agent, and the second one is known to A. Therefore the challenge we face is to privately compute the following sum of products, denoted by $\rho(A,x)$:

$$\rho(A,x) = \sum_{\forall B_i \in S} DTE(B_i,x)\cdot TM(A,B_i) = \sum_{\forall B_i \in S} ITE(A,B_i,x)$$

where S denotes the trust-set of A.

We proceed to presenting three different schemes to compute this sum. In all three schemes we assume a *semi-honest* protocol. That is, members follow the protocol honestly, but may remember information and use it for their own benefit. In section 3.1, we discuss the case of dishonest users.

Scheme 1: Trust relations aware

The first scheme, presented in Algorithm 1, assumes that every member $B_i \in S$ knows the trust value, $TM(A,B_i)$, that member A has in her, or that A is willing to disclose this information to B_i. It is also assumed that a partially trusted third party Z exist (i.e. Z does not collude with other agents, and she has no access to any private information of the involved parties).

The encryption ensures that only B_i knows $ITE(A,B_i,x)$. The permutation carried out by Z ensures that when the set of values $ITE(A,B_i,x)$ is received by

Algorithm 1 Trust relations aware

1: A sends around her public key K_A to all $B_i \in S$ along with $TM(A, B_i)$
2: Each member B_i computes $ITE(A, B_i, x)$, encrypts it and sends the encrypted form $C(K_A, ITE(A, B_i, x))$ to the third party Z.
3: Z generates a random permutation of the encrypted messages and sends it to A.
4: A decrypts the messages and computes the required sum $\rho(A, x)$.

A, the origin of the $ITE(A, B_i, x)$ value is not clear to A. This scheme achieves our computational goal and therefore enables private computation of reputation by any member A. Collusion between any of the members (excluding Z) in this scheme is not helpful, since they have no access to information related to non-colluding members. The main advantage of this scheme is its simplicity and small communication overhead (basically $O(n)$), while its main disadvantage is the disclosure of member trust values. Although deemed otherwise, this privacy violation is not very severe: being in A's trust-set, members already know that their corresponding trust value is above the threshold α.

Scheme 2: Trusted third party dependent

In this scheme, depicted in Algorithm 2, we try to overcome the major disadvantage of the former method and assume that members do not know the trust other members have in them. Unlike our previous scheme, only a trusted third party Z is entrusted with the values of the trust A has in the different members, i.e. Z knows (or A sends it) all the values of $TM(A, B_i)$.

Algorithm 2 Trusted third party dependent

1: A sends around her public key K_A to all $B_i \in S$.
2: Each member B_i computes $C(K_A, DTE(B_i, X))$ and sends it to Z.
3: Z generates a random permutation of all the values received and sends a vector of values to A.
4: A decrypts the encrypted vector of trust in expert values DTE.
5: A sends the vector TM of values $TM(A, B_i)$ to Z.
6: Z permutes the vector with the same permutation used in step (3).
7: A and Z compute the scalar product of DTE and TM vectors using secure computation and obtain the desired sum of products $\rho(A, X)$ (cf. [2] for one possible method to compute scalar product in a secure way).

The main advantage of this method over the previous one, is that individual members do not know A's trust in them. On the other hand the third party Z needs to be trusted with these private values. Moreover, the communication overhead of this scheme is higher, because of the use of secure scalar product, which is an expensive operation (using [2] the complexity is linear but requires the *Add Vectors* protocol which involves many cryptographic transformations).

Scheme 3: Controlled sequence

This scheme, depicted in Algorithm 3, uses the additive scheme described in [9] which enables computing the sum of trust values in an expert, in a private way, independent of any trusted third party. Let us denote the sum of trust values, members in S have in x, as $\tau(S, x)$:

$$\tau(S, x) = \sum_{\forall B_i \in S} DTE(B_i, x)$$

Algorithm 3 Controlled sequence

1: A computes $\tau(S, X)$ using one of the three algorithms described in [9].
2: A decides on a random permutation of members $B_i \in S$ and informs each B_i of the agent following her (next in line) in the permutation.
3: A sends to B_i: $TM'(A, B_i) = (TM(A, B_i) + Q)$, where Q is a random number. We denote $ITE'(A, B_i, x) = TM'(A, B_i) \cdot DTE(Bi, x)$, for simplicity.
4: Using its partial knowledge of A's permutation, B_1 sends the value $ITE'(A, B_1, x)$ to B_2. B_2 sends $ITE'(A, B_1, x) + ITE'(A, B_2, x)$ to B_3. B_3 repeats this process and so do, all the following agents.
The last member sends A the sum $\sum_{\forall B_i \in S} ITE'(A, B_i, x)$.
5: A subtracts $Q \cdot \tau(S, x)$ from the sum it received from the last member, and obtains $\rho(A, x)$ - the desired value.

$$\begin{aligned}\rho(A, x) &= \sum_{\forall B_i \in S} ITE(A, B_i, x) = \sum_{\forall B_i \in S} TM(A, B_i) \cdot DTE(Bi, x) \\ &= \sum_{\forall B_i \in S} (TM(A, B_i) + Q) \cdot DTE(Bi, x) - Q \cdot \sum_{\forall B_i \in S} DTE(B_i, x) \\ &= \sum_{\forall B_i \in S} ITE'(A, B_i, x) - Q \cdot \tau(S, x)\end{aligned}$$

This scheme has the advantage that no trusted third party is involved in it, and it therefore provides more privacy in that respect. However, it has several disadvantages. First, since A is selecting the members it sends the TM' values to, it may select a very small group (e.g. two) thus reducing privacy considerably. This can be overcome by assuring a minimal size to the set of trusted members as discussed in [9] (i.e the witness set). Another way of achieving at least size n group is by having the members use a secret-sharing scheme of at least size n [13], where one of the members must be able to solve the secret. A second disadvantage is that the trust of A in other members, which is a private value, may be compromised by collusion. Any two colluding members $B_i, B_j, i, j \in S$ knowing an approximate value of A's trust in them may be able to approximate Q and find the difference between their respective $TM(A, B_i), TM(A, B_j)$ by a simple subtraction. A way to solve it is to divide the trust set S into two (or more) subsets, which only A knows their composition and use different random numbers for each such subset, again making sure the subset is large enough. Another idea is the following: Let A run the protocol twice. Once with a single Q value, and in the second run, with many different random Qs. Since the users don't know which time is the correct one, their guessing probability is

reduced to 50% (and can be reduced further by re-running the protocol for any arbitrary number of times). This can reduce considerably the risk of disclosing $TM(A, B_j)$ values. The third disadvantage is that the communication overhead in this scheme is even higher than in the second scheme since the complexity of computing $\tau(S, x)$ is $O(n^3)$(see [9]).

3.1 Analysis of dishonest users

In the three schemes presented, we consider two forms of dishonesty: the first includes attempts to compromise private data, and the second includes attempts to bias trust values (malicious behavior). A's motivation to act dishonestly is to learn private information (e.g. a member's trust in an expert). Members of A's trust set act dishonestly when they try to find out A's trust in them or if they attempt to bias an expert reputation by providing wrong input.

Members in S gain from acting dishonestly only when A's identity is known to them. We assume that when this information is kept private the agents will follow the above protocols (although members may be strongly biased toward / against expert x, this bias is accepted by A and affects A's trust value in B_i).

In the first scheme members know exactly who the requester A is. If they want to mislead A, they can promote or demote the value $ITE(A, B_i, x)$ they provide to the trusted third party, Z. Also, since each member participating in scheme 1 knows the trust value of A in them, no misconduct is expected in an attempt to infer $TM(A, B)$. However, in this scheme, members may also attempt to gather information about the nature of the relation other members have with A. This is not possible if we assume secure communication, as A sends its information directly to each member, and these only sends their information to Z. In the second scheme, A's identity can be protected. Thus, members have no motivation to mislead A since they don't know who A is. In the third scheme only the first and last users know who A is and A can select these as trusted friends, the rest have no motivation to be malicious.

The requester A may act dishonestly in order to learn a member's direct trust in x by setting the member trust values accordingly (even if we assume that TM of zero is not a valid value). For example, if the trust in a member is a value in the range 1..10 and the trust in an expert is a value in the range 1..5. A may guess the exact trust in an expert by setting one TM value to 1 and the rest to 6. In the first scheme A cannot fake her trust in members of her trust set since we assume that they are aware of A's trust in them. In the second scheme we can use Z to examine the sets of different $TM(A, B_i)$ values that are either suspicious or those that allow A to infer the trust a member has in x with good probability. In the third scheme we can reduce the risk of A faking her trust in members by a verification procedure carried out by members of the trust set that detect very low values as suspicious.

4 Conclusions

In this paper we discussed the problem of computing reputation of members in a community while preserving the privacy of sensitive information. Such informa-

tion includes the rating of individual members and the trust that one member has in another. The paper presents three schemes for privacy preserving computation and analyzes their privacy characteristics and communication overhead. The presented schemes apply techniques for secure summation [9] and dot product [2] and use these as primitives in a virtual community oriented setting. Our constructs extends state of the art work to elevate existing trust based models in which privacy is a major concern.

Although the schemes were presented in the context of a specific trust and reputation model, the Knots model, they may be used by other models which take into account the same sensitive information (members ratings and members trust). Future work will include formal proof of the amount of privacy provided, an implemented architecture of the suggested protocols and further discussion of privacy issues in trust and reputation systems.

References

1. eBay, http://www.ebay.com/.
2. A. Amirbekyan and V. Estivill-Castro. A new efficient privacy-preserving scalar product protocol, 2007.
3. S. Chakraborty and I. Ray. Trustbac: integrating trust relationships into the rbac model for access control in open systems. In *Proceedings of the eleventh ACM symposium on Access control models and technologies (SACMAT '06)*, pages 49–58, New York, NY, USA, 2006. ACM.
4. N. Gal-Oz, T. Grinshpoun, E. Gudes, and A. Meisels. Cross-community reputation: Policies and alternatives, 2008.
5. N. Gal-Oz, E. Gudes, and D. Hendler. A robust and knot-aware trust-based reputation model, 2008.
6. J. Y.-J. Hsu, K.-J. Lin, T.-H. Chang, C.-J. Ho, H.-S. Huang, and W.-R. Jih. Parameter learning of personalized trust models in broker-based distributed trust management. *Information Systems Frontiers*, 8(4):321–333, 2006.
7. A. Josang and R. Ismail. The beta reputation system, 2002.
8. L. Mui, M. Mohtashemi, and A. Halberstadt. A computational model of trust and reputation for e-businesses, 2002.
9. E. Pavlov, J. S. Rosenschein, and Z. Topol. Supporting privacy in decentralized additive reputation systems, 2004.
10. T. P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing, 1991.
11. F. Pingel and S. Steinbrecher. Multilateral secure cross-community reputation systems for internet communities, 2008.
12. P. Resnick and R. Zeckhauser. Trust among strangers in Internet transactions: Empirical analysis of eBay's reputation system. In *The Economics of the Internet and E-Commerce*.
13. A. Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.
14. S. Steinbrecher. Design options for privacy-respecting reputation systems within centralised internet communities, 2006.
15. B. Yu and M. Singh. Detecting deception in reputation management, 2003.