k-Shares: A Privacy Preserving Reputation Protocol for Decentralized Environments

Omar Hasan[1], Lionel Brunie[1], and Elisa Bertino[2]

[1] INSA Lyon, France
{omar.hasan,lionel.brunie}@insa-lyon.fr
[2] Purdue University, IN, USA
bertino@cs.purdue.edu

Abstract. A reputation protocol computes the reputation of an entity by aggregating the feedback provided by other entities in the system. Reputation makes entities accountable for their behavior. Honest feedback is clearly a pre-requisite for accurate reputation scores. However, it has been observed that entities often hesitate in providing honest feedback, mainly due to the fear of retaliation. We present a privacy preserving reputation protocol which enables entities to provide feedback in a private and thus uninhibited manner. The protocol, termed k-shares, is oriented for decentralized environments. The protocol has linear message complexity under the semi-honest adversarial model, which is an improvement over comparable reputation protocols.

1 Introduction

In recent years, reputation systems have gained popularity as a solution for securing distributed applications from misuse by dishonest entities. A reputation system computes the reputation scores of the entities in the system based on the feedback provided by fellow entities. A popular reputation system is the eBay reputation system (ebay.com), which is used to discourage fraudulent activities in e-commerce. Other well-known examples include the EigenTrust [1] reputation system and the Advogato.org reputation system [2].

An accurate reputation score is possible only if the feedbacks are accurate. However, it has been observed that the users of a reputation system may avoid providing honest feedback [3]. The reasons for such behavior include fear of retaliation from the target entity or mutual understanding that a feedback value would be reciprocated.

A solution to the problem of lack of honest feedback is computing reputation scores in a privacy preserving manner. A privacy preserving protocol for computing reputation scores operates such that the individual feedback of any entity is not revealed. The implication of private feedback is that there are no consequences for the feedback provider and thus he is uninhibited to provide honest feedback.

In this article, we are interested in developing a privacy preserving reputation protocol that is decentralized and efficient. Our motivation stems from the observation that there are currently few if any efficient privacy preserving reputation protocols for decentralized environments, which include peer-to-peer networks, MANETs, and decentralized social networks, such as FOAF (foaf-project.org). The reader is referred to section 7 for related work.

We propose the k-Shares protocol, which is decentralized as well as efficient (linear message complexity). The protocol is shown to be secure under the semi-honest adversarial model. Extensions for security under the malicious model are also discussed.

2 General Framework

2.1 Agents, Trust, and Reputation

We model our environment as a multi-agent environment. Set A is defined as the set of all agents in the environment. $|A| = N$. We subscribe to the definition of trust by sociologist Diego Gambetta [4], which is one of the most commonly accepted definitions of trust. Our formal definition captures the characteristics of trust identified in Gambetta's definition, which include: 1) Binary-Relational and Directional, 2) Contextual, and 3) Quantifiable as Subjective Probability.

Definition 1. ***Trust.*** *Let the trust of a truster a in a trustee b be defined as the tuple: $\langle aTb, \psi, P(perform(a, b, \psi) = true)\rangle$. T is a binary relation on the set A. aTb implies that a has a trust relationship with b or simply that a trusts b. The binary relation T is non-reflexive, asymmetric, and non-transitive. The context of a truster a's trust in a trustee b is an action ψ that the truster a anticipates the trustee b to perform. The set of all actions is given as Ψ. The subjective probability $P(perform((a, b), \psi) = true)$ is the quantification of a truster a's trust in a trustee b. $perform((a, b), \psi)$ is a function, such that: $perform : T \times \Psi \rightarrow \{true, false\}$. $perform$ outputs $true$ if the trustee b does in fact perform the action anticipated by the truster a. On the contrary, if the trustee does not perform the anticipated action, the function outputs $false$. When the context ψ is clear, $l_{ab} \equiv P(perform((a, b), \psi) = true)$.*

Some examples of actions: "prescribe correct medicine", "repair car", "deliver product sold online", etc.

Definition 2. ***Source Agent.*** *An agent a is said to be a source agent of an agent b in the context of an action ψ if a has trust in b in context ψ. In other words, agent a is a source agent of agent b in context ψ if trust $\langle aTb, \psi, P(perform(a, b, \psi) = true)\rangle$ exists. The set of all source agents of an agent b in context ψ is given as $S_{b,\psi} = \{a \mid \langle aTb, \psi, P(perform(a, b, \psi) = true)\rangle$ exists$\}$. When the context is clear, the notation S_b may be used instead of $S_{b,\psi}$. The quantification of a source agent a's trust in agent b is referred to as feedback.*

Definition 3. ***Reputation.*** *Let $S_t = \{a_1 \ldots a_n\}$ be the set of source agents of an agent t in context ψ. This implies that each agent $a \in S_t$ has the trust $\langle aTt, \psi, P(perform(a, t, \psi) = true)\rangle$ in agent t. Then the reputation of agent t in context ψ is given as the function: $rep(P(perform(a_1, t, \psi) = true), \ldots,$*

$P(perform(a_n, t, \psi) = true))$, or in simpler notation: $rep(l_{a_1 t}, \ldots, l_{a_n t})$, such that: $rep : [0,1]_1 \times \ldots \times [0,1]_n \rightarrow \mathbb{R}$. The reputation of an agent t is represented by the variable $r_{t,\psi}$, or r_t when the context is clear.

Definition 4. ***Function*** $rep_{\oplus}$***.*** *Let function $rep_{\oplus}$ be a realization of the function rep. $rep_{\oplus} : [0,1]_1 \times \ldots \times [0,1]_n \rightarrow [0,1]$. $rep_{\oplus}$ is implemented as follows ($l_{a_1 t} \ldots l_{a_n t}$ are feedback values in a context ψ): $rep_{\oplus}(l_{a_1 t} \ldots l_{a_n t}) = \frac{l_{a_1 t} + \ldots + l_{a_n t}}{n} = \frac{\sum_{i=1}^{n} l_{a_i t}}{n}$.*

We have defined the reputation of an agent as any function that aggregates the feedback of its source agents. The function $rep_{\oplus}$ implements the reputation of an agent t as the mean of the feedbacks of all its source agents. Our decision to define reputation in such simple but intuitive terms is influenced by the eBay reputation system (ebay.com). The eBay reputation system, which is one of the most successful reputation systems, represents reputation as the simple sum of all feedbacks. We go one step further and derive the average from the sum in order to normalize the reputation values. Please note that $rep_{\oplus}$ is a function constructed from summation. With summation, it is possible to model reputation as any function that can be approximated as a polynomial expression.

Definition 5. ***Reputation Protocol.*** *Let Π be a multi-party protocol. Then Π is a Reputation Protocol, if 1) the participants of the protocol include: a querying agent q, a target agent t and all n source agents of t in a context ψ, 2) the inputs include: the feedbacks of the source agents in context ψ, and 3) the output of the protocol is: agent q learns the reputation $r_{t,\psi}$ of agent t.*

2.2 Adversary

We refer to the coalition of dishonest agents as the adversary. Adversarial models:

Semi-Honest. In the semi-honest (honest-but-curious) model, the agents always execute the protocol according to specification. The adversary abstains from wiretapping and tampering of the communication channels. However, within these constraints, the adversary passively attempts to learn the inputs of honest agents by using intermediate information received during the protocol.

Malicious. Malicious agents are not bound to conform to the protocol. They may attempt to learn private inputs as well as to disrupt the protocol for honest agents. The reasons for disrupting the protocol may range from gaining illegitimate advantage over honest agents to completely denying the service of the protocol to honest agents.

In this paper, we propose a solution for the first model. Ideas for an efficient solution for the second model are discussed in section 5.

2.3 Privacy

Definition 6. ***Private Data.*** *Let x be some data and an agent a be the owner of x. Then x is agent a's private data if agent a desires that no other agent learns x. An exception is those agents to whom a reveals x herself. However, if a reveals x to an agent b, then a desires that b does not use x to infer more information. Moreover, a desires that b does not reveal x to any third party.*

Definition 7. ***Preserving Privacy (by an Agent).*** *Let x be an agent a's private data. Then an agent b is said to preserve the privacy of agent a, if 1) a reveals x to b, 2) b does not use x to infer more information, and 3) b does not reveal x to any third party. We define action $\rho =$ "preserve privacy".*

The action "preserve privacy" is synonymous with the action "be honest", since an agent preserves privacy only if it is honest, and an honest agent always preserves privacy since it has no ulterior motives.

Definition 8. ***Trusted Third Party (TTP).*** *Let $S \subseteq A$ be a set of n agents, and $TTP_S \in A$ be an agent. Then TTP_S is a Trusted Third Party (TTP) for the set of agents S if for each $a \in S$, $P(perform(a, TTP_S, \rho) = true) = 1$.*

We adopt the Ideal-Real approach [5] to formalize the privacy preservation property of a protocol. In this article we use the term *high* as a probability variable that may be realized to a specific value according to the security needs of an application. For example, in the experiments (section 6) on the Advogato.org web of trust, we consider *high* probability as 0.90. Consequently, low probability is the complement of high probability.

Definition 9. ***Ideal Privacy Preserving Reputation Protocol.*** *Let Π be a reputation protocol, which includes as participants: a querying agent q, a target agent t, and $S_t = S_{t,\psi}$, the set of all n source agents of t in context ψ. Then Π is an ideal privacy preserving reputation protocol under a given adversarial model, if: 1) the inputs of all n source agents of t are private; 2) TTP_{S_t} is also a participant; 3) $m < n$ of the source agents (given as set M) and agents q and t are considered to be dishonest, however, q wishes to learn the correct output; 4) agents $S_t - M$ and TTP_{S_t} are honest; 5) as part of the protocol, TTP_{S_t} receives the private inputs from the source agents and outputs the reputation $r_{t,\psi}$ to agent q; and 6) over the course of the protocol, the private input of each agent $a \in S_t - M$ is revealed only to TTP_{S_t}.*

In an ideal privacy preserving reputation protocol, it is assumed that for each agent $a \in S_t - M$, the adversary does not gain any more information about the private input of agent a from the protocol other than what it can deduce from what it knows before the execution of the protocol and the output, with probability $P(perform(a, TTP_{S_t}, \rho) = true)$, under the given adversarial model.

Definition 10. ***Real Privacy Preserving Reputation Protocol.*** *Let $\mathcal{I}$ be an ideal privacy preserving reputation protocol. Then $\mathcal{R}$ is a real privacy preserving reputation protocol under a given adversarial model, if: 1) $\mathcal{R}$ has the same* *parameters (participants, private inputs, output, adversary, honest agents, setup, etc.) as $\mathcal{I}$, except that there is no TTP_{S_t} as a participant; and 2) the adversary learns no more information about the private input of an agent a than it learns in protocol $\mathcal{I}$, with high probability, when both protocols are operating under the given adversarial model.*

3 Problem Definition

Definition 11. ***Problem Definition.*** *Let $S_{t,\psi} = \{a_1 \dots a_n\}$ be the set of all source agents of agent t in the context of action ψ. Find a reputation protocol Π, which takes private input $l_{at} \equiv P(perform(a, t, \psi) = true)$ from each agent $a \in S_t$, and outputs the reputation $r_{t,\psi}$ of the target agent t to a querying agent q. Reputation is realized as $rep_{\oplus}$. Agents q, t, and $m < n$ of the source agents are considered to be dishonest, however, q wishes to learn the correct output. The reputation protocol Π is required to be decentralized and secure under the semi-honest model.*

4 The k-Shares Reputation Protocol

In this section we present our k-shares protocol, which is a real privacy preserving reputation protocol under the semi-honest model. The k-shares protocol is inspired by a protocol in [6] (section 5.2). However, our protocol has a lower message complexity of only $O(kn)$ as opposed to the complexity of $O(n^2)$ of the protocol in [6]. In the experiments we observe that k can be set as low as 2, while preserving the privacy of a high majority of agents. Moreover, the extended version of our protocol allows agents to abstain when their privacy is not assured. The important steps of the protocol are outlined below.

1. **Initiate.** The protocol is initiated by a querying agent q to determine the reputation $r_{t,\psi}$ of a target agent t. Agent q retrieves $S_t \equiv S_{t,\psi}$, the set of source agents of agent t in context ψ. Agent q then sends S_t to each agent $a \in S_t$.
2. **Select Trustworthy Agents.** Each agent $a \in S_t$ selects upto k other agents in S_t. Let's refer to these agents selected by a as the set $U_a = \{u_{a,1} \dots u_{a,k_a}\}$, where $1 \leq k_a \leq k$. Agent a selects these agents such that: $P(perform(a, u_{a,1}, \rho) = false) \times \dots \times P(perform(a, u_{a,k_a}, \rho) = false)$ is low. That is, the probability that all of the selected agents will collude to break agent a's privacy is low.
3. **Prepare Shares.** Agent a then prepares $k_a + 1$ shares of its secret feedback value l_{at}. The shares, given as: $x_{a,1} \dots x_{a,k_a+1}$, are prepared as follows: The first k_a shares are random numbers uniformly distributed over a large interval. The last share is selected such that: $\sum_{i=1}^{k_a+1} x_{a,i} = l_{at}$. That is, the sum of the shares is equal to the feedback value. Since each of the $k_a + 1$ shares is a number uniformly distributed over a large interval, no information about the secret can be learnt unless all of the shares are known.

4. **Send Shares.** Agent a sends the set $U_a = \{u_{a,1} \ldots u_{a,k_a}\}$ to agent q. Agent a sends $x_{a,i}$ to agent $u_{a,i}$, where $i \in \{1 \ldots k_a\}$.
5. **Receive Shares.** Agent q receives U_a from each agent $a \in S_t$. Then for each agent a, agent q: 1) compiles the list of agents from whom a should expect to receive shares, and 2) sends this list to agent a. Agent a then proceeds to receive shares from the agents on the list provided by q.
6. **Compute Sums.** Agent a computes σ_a, the sum all shares received and its own final share x_{a,k_a+1}. Agent a sends the sum σ_a to q.
7. **Compute Reputation.** Agent q receives the sum σ_a from each agent $a \in S_t$. q computes $r_{t,\psi} = (\sum_{a \in S_t} \sigma_a)/n$.

4.1 Protocol Specification

The protocol is specified in Figure 1. The function $set\_of\_trustworthy(a, S)$ returns a set of agents $U_a = \{u_{a,1} \ldots u_{a,k_a}\}$, where $1 \leq k_a \leq k$, and $U_a \subseteq S$. The set U_a is selected such that: $P(perform(a, u_{a,1}, \rho) = false) \times \ldots \times P(perform(a, u_{a,k_a}, \rho) = false)$ is low, with the minimum possible k_a.

4.2 Security Analysis

Correctness. Each agent $a \in S_t$ prepares the shares $x_{a,1} \ldots x_{a,k_a+1}$ of its feedback value l_{at}, such that: $\sum_{j=1}^{k_a+1} x_{a,j} = l_{at}$. The sum of the feedback values of all agents in $S_t = \{a_1 \ldots a_n\}$ is given as: $\sum_{i=1}^{n} l_{a_i t}$. Thus, the sum of the feedback values of all agents in S_t can be stated as: $\sum_{i=1}^{n}(\sum_{j=1}^{k_{a_i}+1} x_{a_i,j})$. That is, the sum of all shares of all agents.

Each agent $a \in S_t$ provides agent q the set U_a, which is the set of agents whom a is going to send its shares. After q has received this set from all agents in S_t, it compiles and sends to each agent a, the set J_a, which is the set of agents who are in the process of sending a share to agent a. Thus, each agent a knows exactly which and how many agents, it will receive a share from. When agent a has received all of those shares, it sends σ_a, the sum of all shares received and its final share, to agent q. Previously, each agent $a \in S_t$ sends each of his shares $x_{a,1} \ldots x_{a,k_a}$, once to only one other agent, and adds the final share x_{a,k_a+1} once to his own σ_a. It follows that the sums $\sigma_{a_1} \ldots \sigma_{a_n}$ include all shares of all agents and that they include each share only once.

The final value of r in the protocol is: $r = \sum_{i=1}^{n} \sigma_{a_i} = \sum_{i=1}^{n}(\sum_{j=1}^{k_{a_i}+1} x_{a_i,j}) = \sum_{i=1}^{n} l_{a_i t}$. Thus when q computes $r_{t,\psi} = r/n$, it is the correct reputation of agent t in context ψ (Definition 3).

Privacy. Let's consider an agent $a \in S_t$. Agent a prepares the shares $x_{a,1} \ldots x_{a,k_a+1}$ of its secret feedback value l_{at}. The first k_a shares $x_{a,1} \ldots x_{a,k_a}$ are random numbers uniformly distributed over a large interval $[-X, X]$. The final share, $x_{a,k_a+1} = l_{at} - \sum_{i=1}^{k_a} x_{a,i}$, is also a number uniformly distributed over a large interval since it is a function of the first k_a shares which are random numbers. Thus, individually each of the shares does not reveal any information about the secret feedback value l_{at}. Moreover, no information is learnt about l_{at} even if upto k_a shares are known, since there sum would be some random number uniformly distributed over a large interval. The only case in which information can be gained about l_{at} is if all $k_a + 1$ shares are known. Then, $l_{at} = \sum_{i=1}^{k_a+1} x_{a,i}$.

We now analyze if the $k_a + 1$ shares of an agent a can be learnt by the adversary from the protocol.

Agent a sends each share $x_{a,i}$ only to agent $u_{a,i}$, where $i \in \{1 \ldots k_a\}$. Each $u_{a,i}$ then computes $\sigma_{u_{a,i}}$, which is the sum of all shares that it receives and its own final share $x_{u_{a,i},k_{u_{a,i}}+1}$. Even if agent a is the only agent to send agent $u_{a,i}$ a share, $\sigma_{u_{a,i}} = x_{a,i} + x_{u_{a,i},k_{u_{a,i}}+1}$. That is, the sum of agent a's share and agent $u_{a,i}$'s final share. $\sigma_{u_{a,i}}$ is a number uniformly distributed over a large interval. Thus, when agent $u_{a,i}$ sends this number to agent q, it is impossible for q to distinguish the individual shares from the number. Therefore, each share $x_{a,i}$ that agent a sends to agent $u_{a,i}$ will only be known to agent $u_{a,i}$. Unless, agent $u_{a,i}$ is dishonest. The probability that agent $u_{a,i}$ is dishonest, that is, it will attempt to breach agent a's privacy is given as: $P(perform(a, u_{a,i}, \rho) = false)$.

To learn the first k_a shares of agent a, all agents $u_{a,1} \ldots u_{a,k_a}$ would have to be dishonest. The probability of this scenario is given as: $P(perform(a, u_{a,1}, \rho) = false) \times \ldots \times P(perform(a, u_{a,k_a}, \rho) = false)$.

Even in the above scenario, the adversary does not gain information about l_{at}, without the knowledge of agent a's final share x_{a,k_a+1}. However, agent a has to send $\sigma_a = x_{a,k_a+1} + \sum_{v \in J_a} x_v$, and agent a has no control over the $\sum_{v \in J_a} x_v$ portion of the equation. Therefore, we assume that agent q learns the final share of agent a.

Thus the probability that the protocol will not preserve agent a's privacy can be stated as: $P(perform(a, u_{a,1}, \rho) = false) \times \ldots \times P(perform(a, u_{a,k_a}, \rho) = false)$. If we assume that the agents $u_{a,1} \ldots u_{a,k_a}$ are selected such that this probability is low, then with high probability, the adversary learns no more information about l_{at} than it can learn in an ideal protocol with what it knows before the execution of the protocol and the outcome.

The protocol Semi-Honest-k-Shares is a real privacy preserving reputation protocol (Definition 10) under the semi-honest model, since: 1) Semi-Honest-k-Shares has the same parameters as an ideal protocol (except the TTP), and 2) the adversary learns no more information about the private input of an agent a in Semi-Honest-k-Shares than it learns in an ideal protocol, with high probability, under the semi-honest adversarial model.

4.3 An Extension

The privacy of the k-Shares protocol depends on the assumption that each agent $a \in S_t$ will find trustworthy agents in S_t. However, the protocol may be extended such that agents are allowed to abstain when they don't find trustworthy agents. In that case, an agent would generate two shares whose sum equals zero. One of the shares would be sent to a random source agent and the other to the querying agent along with any shares received added to it. In section 6.2, we observe that the protocol computes sufficiently accurate reputation scores even if a large number of agents abstain.

4.4 Complexity Analysis

The protocol requires $4n + O(kn) + 2$ messages to be exchanged (complexity: $O(n)$). In terms of bandwidth used, the protocol requires transmission of the following amount of information: $n^2+5n+O(n^2)+O(3kn)$ agent IDs (complexity: $O(n^2)$), and $n + O(kn)$ numbers (complexity: $O(n)$).

The protocol requires $4n + O(kn) + 2$ messages to be exchanged (complexity: $O(n)$). In terms of bandwidth used, the protocol requires transmission of the following amount of information: $n^2+5n+O(n^2)+O(3kn)$ agent IDs (complexity: $O(n^2)$), and $n + O(kn)$ numbers (complexity: $O(n)$).

4.5 Discussion

We pose the following question: Why send shares of the secret feedback value to $n - 1$ potentially unknown agents when privacy can be assured by sending shares to only $k < n - 1$ trustworthy agents? In the k-Shares protocol, each agent a relies on at most k agents who are selected based on a's knowledge of their trustworthiness in the context of preserving privacy. This is in contrast to the protocol in [6] which requires each agent to send shares to all other $n - 1$ source agents in the protocol. As we observe in section 6, the privacy of a high majority of agents can be assured with k as small as 2. Moreover, increasing k to values approaching $n - 1$ has no significant advantage.

5 Extensions for the Malicious Model – Future Work

Malicious agents may take the following additional actions: 1) drop messages, 2) add values that are out of range. The solution that we propose for the first problem is to extend the k-Shares protocol as follows: all messages are encrypted with an additive homomorphic cryptosystem, and relayed through the querying agent. Thus, the querying agent would know if an agent has dropped a message. The solution to the second problem is that along with its shares, each source agent provides a zero knowledge proof demonstrating that the sum of the shares lies in the correct range. Wiretapping and tampering may be prevented by securing communication channels with a protocol such as SSL or IPSec. These extensions would raise the computational complexity of the protocol, however the message complexity would remain as $O(n)$. This is in contrast to the protocol for the malicious model in [6] which has a complexity of $O(n^3)$.

6 Experiments

6.1 The Dataset: Advogato.org

We use the real web of trust of Advogato.org [2] as the dataset for our experiments. The members of Advogato rate each other in the context of being active and responsible members of the open source software developer community. The choice of feedback values are *master*, *journeyer*, *apprentice*, and *observer*, with *master* being the highest level in that order. The result of these ratings is a rich web of trust, which comprises of $13,904$ users and $57,114$ trust ratings (November 20, 2009). The distribution of ratings is as follows: *master*: 31.7%, *journeyer*: 40.3%, *apprentice*: 18.7%, and *observer*: 9.3%.

The members of Advogato are expected to not post spam, not attack the Advogato trust metric, etc. Thus we posit that on Advogato, the context "be a responsible member of the open source software developer community", comprises of the context "be honest". Since we quantify trust as probability, we substitute the four feedback values of Advogato as follows: $master = 0.99$, $journeyer = 0.70$, $apprentice = 0.40$, and $observer = 0.10$. These substitutions are made heuristically based on our experience with Advogato.

For the experiments, we define the lowest acceptable probability that privacy will be preserved as 0.90. This means that a set of two trustworthy agents must include either one *master* rated agent or two *journeyer* rated agents for this threshold to be satisfied.

6.2 Experiment 1

Objective: In the protocol Semi-Honest-k-Shares, the following assumption must hold for an agent a's privacy to be preserved: $P(perform(a, u_{a,1}, \rho) = false) \times \ldots \times P(perform(a, u_{a,k_a}, \rho) = false)$ is low. That is, the probability that the agents to whom agent a sends shares, are all dishonest must be low. **We would like to know the percentage of instances of source agents for whom this assumption holds true.**

Algorithm: A randomly selected querying agent queries the reputation of every other agent who has at least min source agents. Over the course of all queries, we observe the probability $P(perform(a, u_{a,1}, \rho) = false) \times \ldots \times P(perform(a, u_{a,k_a}, \rho) = false)$, for each source agent a. The experiment is run for each value of min in $\{5, 10, 15, 20, 25, 50, 75, 100, 500\}$.

Results: For $min = 25$, we observe that the assumption holds for 81.7% of instances of source agents. Additionally, 85.8% for $min = 50$, 87.0% for $min = 75$, 87.4% for $min = 100$, and 87.5% for $min = 500$. We note that the increase in the percentage is significant up to $min = 100$. This is due to the greater choice of trustworthy agents available for each agent when the protocol has more source agents. At $min = 5$, the percentage is 72.5%, which implies that approximately 30% of the source agents will have to abstain. However, in a separate experiment (full details not included due to space limitation), we observed that at $min = 25$, even if only around 40% of agents participate, over 95% of the computed reputation scores have an error of at most 0.1 compared to the true scores. Additionally, over 85% at $min = 10$, and over 90% at $min = 15$. Thus, even a significant portion of agents abstaining does not pose an issue.

6.3 Experiment 2

Objective: We would like to know the effect of increasing k on the percentage of instances of source agents whose privacy is preserved in the protocol Semi-Honest-k-Shares.

Algorithm: A randomly selected querying agent queries the reputation of every other agent who has at least *min* source agents. We vary k and observe the percentage of instances of source agents whose privacy is preserved. The set of experiments is run with $min = 50$.

Results: For $min = 50$, and $k = 1$, we observe that the percentage is 75.4%, and at $k = 2$, the percentage is 85.8%. The jump is due to the possibility with $k = 2$ to rely on two *journeyer* agents. With $k = 1$, the only possibility is to rely on one *master* agent. However, increasing k over 2, even up to 500, does not result in a significant advantage (86.3% at $k = 500$). Thus, in this dataset, privacy can be preserved for a high percentage of source agents with k as small as 2. This results in a very efficient protocol. This is in contrast to the protocol presented in [6], which requires each agent to send shares to $n-1$ agents, resulting in $O(n^2)$ message complexity.

7 Related Work

The inspiration for the k-Shares protocol comes from [6]. However, among other advantages (section 4.5), our protocol requires $O(kn)$ messages as opposed to the $O(n^2)$ required by [6]. Additionally, we also evaluate our protocols on a large and real dataset.

A number of privacy preserving reputation systems are based on the premise that a trusted hardware module is present at every agent. The systems that fall under this category include [7], [8], [9]. A system by Kinateder et al [10] avoids the hardware modules, however it requires anonymous routing infrastructure at the network level. These systems clearly differ from our approach, which does not mandate specialized platforms.

Several privacy preserving reputation systems have the concept of e-cash as their basis. These systems include [11], [12], [13]. However, these systems either rely on TTPs or centralized constructs, such as the "bank" in [13]. In contrast, our reputation protocols are decentralized.

8 Conclusion

In this article we have presented the k-Shares privacy preserving reputation protocol. A defining characteristic of this protocol is that an agent a himself selects the agents that are critical for preserving its privacy. The selection is based on a's knowledge of the trustworthiness of those agents in the context of preserving privacy, thus a is able to maximize the probability that its privacy will be preserved.

The experiments conducted on the real and large dataset of Advogato.org yield favorable results. It is shown that the k-Shares protocol is able to assure the privacy of a large majority of the source agents. The extended protocol allows agents to abstain from providing feedback when their privacy is at risk.

As analyzed, the protocol has linear message complexity and is thus quite efficient. We designed the k-Shares protocol such that the number of trustworthy agents that each agent can send shares to is limited to k. This design choice is validated by the experiment results, which show that the privacy of a high majority of agents can be assured with k as small as 2. Moreover, increasing k to values approaching $n-1$ has no significant advantage.

In conclusion, the k-shares reputation protocol is decentralized, efficient, provides accurate results, and is either able to preserve the privacy of participants with high probability or otherwise allows them to abstain.

References

1. Kamvar, S.D., Schlosser, M.T., GarciaMolina, H.: The eigentrust algorithm for reputation management in p2p networks. In: Proc. of the 12th Intl. Conf. on World Wide Web (WWW 2003). (2003)
2. Levien, R.: Attack resistant trust metrics. Manuscript, University of California - Berkeley. www.levien.com/thesis/compact.pdf (2002)
3. Resnick, P., Zeckhauser, R.: Trust among strangers in internet transactions. The Economics of the Internet and E-Commerce. Vol. 11 of Advances in Applied Microeconomics (2002) 127–157
4. Gambetta, D.: Can We Trust Trust? In: Trust: Making and Breaking Cooperative Relatioins. Department of Sociology, University of Oxford (2000) 213 – 237
5. Goldreich, O.: The Foundations of Crypto. - Vol. 2. Cambridge Univ. Press (2004)
6. Pavlov, E., Rosenschein, J.S., Topol, Z.: Supporting privacy in decentralized additive reputation systems. In: Proc. of the 2nd Intl. Conf. on Trust Management (iTrust 2004). (2004)
7. Kinateder, M., Pearson, S.: A privacy-enhanced peer-to-peer reputation system. In: Proc. of the 4th Intl. Conf. on E-Commerce and Web Techs. (2003)
8. Voss, M., Heinemann, A., Muhlhauser, M.: A privacy preserving reputation system for mobile info. dissemination networks. In: Proc. of the 1st Intl. Conf. on Security and Privacy for Emerging Areas in Comm. Networks (SECURECOMM). (2005)
9. Bo, Y., Min, Z., Guohuan, L.: A reputation system with privacy and incentive. In: Proc. of the 8th ACIS Intl. Conf. on Soft. Eng., AI, Networking, and Parallel/Distributed Comp. (SNPD'07). (2007)
10. Kinateder, M., Terdic, R., Rothermel, K.: Strong pseudonymous comm. for p2p reputation systems. In: Proc. of the 2005 ACM Symp. on Applied Comp. (2005)
11. Ismail, R., Boyd, C., Josang, A., Russell, S.: Strong privacy in reputation systems. In: Proc. of the 4th Intl. Workshop on Info. Security Apps. (WISA'03). (2004)
12. Ismail, R., Boyd, C., Josang, A., Russell, S.: Private reputation schemes for p2p systems. In: Proc. of the 2nd Intl. Workshop on Security in Info. Systems. (2004)
13. Androulaki, E., Choi, S.G., Bellovin, S.M., Malkin, T.: Reputation systems for anonymous networks. In: Proc. of the 8th Privacy Enhancing Technologies Symp. (PETS 2008). (2008)

Protocol: Semi-Honest-k-Shares

Participants: Agents: q, t, $S_t = S_{t,\psi} = \{a_1 \dots a_n\}$. Agents q, t, and a subset of $S_{t,\psi}$ of size $m < n$ are dishonest, however, q wishes to learn the correct output.

Input: Each source agent a has a private input $l_{at} = P(perform(a, t, \psi) = true)$.

Output: Agent q learns $r_{t,\psi}$, the reputation of agent t in context ψ.

Setup: Each agent a maintains $S_a = S_{a,\psi}$, the set of its source agents in context ψ.

Events and Associated Actions (for an Agent a):

need arises to determine $r_{t,\psi}$

▷ initiate query
send tuple (REQUEST_FOR_SOURCES, ψ) to t
receive tuple (SOURCES, ψ, S_t) from t
for each agent $v \in S_t$
    do $J_v \leftarrow \phi$
$S'_t \leftarrow S_t$
$r \leftarrow 0$
$q \leftarrow a$
$s \leftarrow$ timestamp()
send tuple (PREP, q, t, s, S_t) to each agent $v \in S_t$

tuple (REQUEST_FOR_SOURCES, ψ) **received from agent** q

send tuple (SOURCES, ψ, S_a) to q

tuple (PREP, q, t, s, S_t) **received from agent** q

$I \leftarrow \phi$
$J \leftarrow \phi$
$\sigma_a \leftarrow 0$
$U_a \leftarrow$ set_of_trustworthy(a, $S_t - a$)
$k_a \leftarrow |U_a|$
for $i \leftarrow 1$ **to** k_a
    do $x_{a,i} \leftarrow$ random($-X, X$)
$x_{a,k_a+1} \leftarrow l_{at} - \sum_{i=1}^{k_a} x_{a,i}$
send tuple (RECIPIENTS, q, t, s, U_a) to agent q
for each agent $u_{a,i} \in U_a = \{u_{a,1} \dots u_{a,k_a}\}$
    do send tuple (SHARE, $q, t, s, x_{a,i}$) to agent $u_{a,i}$

tuple (RECIPIENTS, q, t, s, U_v) **received from an agent** $v \in S_t$

for each agent $u \in U_v$
    do $J_u \leftarrow J_u \cup v$
$S'_t \leftarrow S'_t - v$
if $S'_t = \phi$
  then $S'_t \leftarrow S_t$
    for each agent $w \in S_t$
      do send tuple (SENDERS, q, t, s, J_w) to agent w

tuple (SHARE, q, t, s, x_v) **received from an agent** $v \in S_t$

$I \leftarrow I \cup v$
$\sigma_a \leftarrow \sigma_a + x_v$
if $I = J$
  then $\sigma_a \leftarrow \sigma_a + x_{a,k_a+1}$
    send tuple (SUM, q, t, s, σ_a) to agent q

tuple (SENDERS, q, t, s, J_a) **received from agent** q

$J \leftarrow J_a$
if $I = J$
  then $\sigma_a \leftarrow \sigma_a + x_{a,k_a+1}$
    send tuple (SUM, q, t, s, σ_a) to agent q

tuple (SUM, q, t, s, σ_v) **received from an agent** $v \in S_t$

$S'_t \leftarrow S'_t - v$
$r \leftarrow r + \sigma_v$
if $S'_t = \phi$
  then $r_{t,\psi} \leftarrow r/n$

Fig. 1. Protocol: Semi-Honest-k-Shares