A Privacy Preserving Reputation Protocol for Web Service Provider Selection

Omar Hasan[1], Lionel Brunie[1], Ernesto Damiani[2]

[1]*CNRS, INSA-Lyon, University of Lyon, F-69621, France*
{*omar.hasan, lionel.brunie*}*@insa-lyon.fr*

[2]*Department of Computer Technology, University of Milan, Italy*
ernesto.damiani@unimi.it

***Abstract*—Reputation systems offer web service consumers an effective solution for selecting web service providers who meet their Quality of Service (QoS) expectations. A reputation system computes the reputation of a provider as an aggregate of the feedback submitted by consumers. Truthful feedback is clearly a pre-requisite for accurate reputation scores. However, it has been observed that users of a reputation system often hesitate in providing truthful feedback, mainly due to the fear of reprisal from target entities. We present a privacy preserving reputation protocol that enables web service consumers to provide feedback about web service providers in a private and thus uninhibited manner.**

I. Introduction

Web services is a technology that provides the means to make the functionality of an application available over the Internet such that it can be used by remote applications. A key advantage of the web services technology is that it enables heterogeneous applications to interact with each other over the Internet regardless of their development and operational platforms. This is possible because the web services technology is defined as a set of standards, notably SOAP (Simple Object Access Protocol), WSDL (Web Services Description Language), and UDDI (Universal Description Discovery Integration), which are all XML-based. XML (EXtensible Markup Language) is a universally accepted and platform independent standard for information representation. Another key feature of the web services technology is that web service applications are discoverable and have self-describing interfaces. This implies that a web service application can be discovered and invoked at runtime by remote applications.

One of the main concerns for a web service consumer when selecting a web service provider is the Quality of Service (QoS) that it will provide. QoS parameters of web services include security, reliability, and performance. The functionality of a web service application (also called web service) is published and is therefore fully known by the consumer. However, the same is not true for the QoS, which may vary drastically from one service provider to another. Thus a web service consumer is faced with a decision about which web service provider to select when a number of them provide web services with similar functionality. A web service provider may offer a Service Level Agreement (SLA), however there are no guarantees that the provider would honor its terms.

In recent years, reputation systems have gained popularity as a solution for determining the trustworthiness of entities in distributed systems. A reputation system computes the reputation score of an entity in the system as an aggregate of the feedback provided by fellow entities. A popular reputation system is the eBay reputation system (ebay.com), which identifies the quality of service provided by sellers in the context of e-commerce.

A number of reputation systems have also been proposed for determining the trustworthiness of web service providers in terms of the quality of service that they claim. Web service consumers grade the quality of service that they receive from web service providers. A potential consumer can query the reputation scores of the providers when faced with the decision of selecting one of them. A provider with high reputation provides high quality of service or otherwise risks losing its reputation and its customer base.

Wang and Vassileva [1] have presented a survey of the reputation systems proposed for selecting web services and web service providers. They broadly classify these systems as centralized vs. decentralized. Centralized reputation systems for web services, such as [2], [3], [4], [5], [6], [7], [8], rely on a central QoS registry that collects and stores QoS data and opinions from web service consumers. In contrast, decentralized approaches such as [9], [10], [11] do away with the central registry and task the consumers themselves with maintaining relevant information.

It is evident that a reputation score will be a true reflection of a web service provider's trustworthiness only if the feedback provided by the web service consumers is truthful. However, it has been observed that the users of a reputation system may avoid providing truthful feedback [12], in particular feedback that is negative. One of the main reasons for such behavior is the fear of reprisal from the target entity [12].

A solution to the problem of lack of truthful 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 truthful feedback.

None of the approaches covered in the survey by Wang and Vassileva [1] (in particular the ones that are decentralized) attempt to preserve the privacy of the opinions of the consumers. We have also not found any privacy preserving reputation systems for web service provider selection proposed after the publication of the noted survey. Our contribution in this article is a privacy preserving reputation protocol for selecting web service providers. The protocol is decentralized, which prevents single points of failure and is in keeping with the heterogeneity and openness of the web services architecture. Moreover, the protocol is efficient. It requires an exchange of $O(n)$ number of messages, where n is the number of web service consumers who provide feedback about the QoS of the target web service provider.

II. Framework

A. Web Service Entities

An entity is a person, an organization, or a software agent that represents a person or an organization. A web service consumer is an entity that consumes web services. A web service provider is an entity that provides web services. Let $\mathbb{C}$ be the set of all web service consumers and let $\mathbb{P}$ be the set of all web service providers in the environment. Let $\mathbb{E}$ be the set of all consumer and provider entities, that is $\mathbb{E} = \mathbb{C} \cup \mathbb{P}$.

B. Actions

Let Ψ_{sp} denote a set of actions that a web service provider can perform in the context of providing web services with certain qualities. Some examples of these actions: "provide web services that are highly available", "provide web services that are secure", "provide web services at the committed QoS over 99% of the time", etc.

Let $\Psi_{sc} = \{\rho\}$ be a set that contains the action $\rho =$ "preserve privacy of fellow entities". The action ρ can be performed by web service consumers to protect the privacy of the opinions of fellow consumers about web service providers.

Let $l_{a,b,\psi}$ represent the subjective probability from an entity a's *local* perspective that an entity b will perform an action ψ. For example, a web service consumer a may believe that a web service provider b will perform an action $\psi =$ "provide web services at the committed QoS over 99% of the time". $l_{a,b,\psi} \equiv l_{a,b}$ when the context ψ is clear.

C. Trust

We subscribe to the definition of trust by sociologist Diego Gambetta. The reason for this choice is the ability to quantify trust as probability, which allows us to quantify the security guarantees of the protocol that we build. The definition is given as follows [13]:

"Trust (or, symmetrically, distrust) is a particular level of the subjective probability with which an agent assesses that another agent or group of agents will perform a particular action, both before he can monitor such action (or independently of his capacity ever to be able to monitor it) and in a context in which it affects his own action."

We infer from Gambetta's definition that trust is binary-relational, directional, contextual, and quantifiable as subjective probability. We now formalize the notion of trust based on these characteristics.

Let $\mathbb{U} = \mathbb{E} \times \mathbb{E}$ be an asymmetric binary relation. Let $\mathbb{T} \subseteq \mathbb{U}$ be the set of all existing trust relationships between entities. $(a, b) \in \mathbb{T}$, where $a, b \in \mathbb{E}$, implies that an entity a has a trust relationship towards an entity b. The asymmetric binary relation $\mathbb{U}$ captures the binary-relational and directional characteristics of trust.

The trust of an entity a in an entity b in the context of an action ψ is given as the tuple $\langle (a, b) \in \mathbb{T}, \psi, l_{a,b,\psi} \rangle$. The action ψ is the context of a's trust in b. The subjective probability $l_{a,b,\psi}$ is the quantification of a's trust in b in the context of an action ψ. We also refer to the quantification of trust in this paper as feedback and opinion.

We establish the following constraints:

If $(a, b) \in \mathbb{T}$, then $a \in \mathbb{C}$ and $b \in \mathbb{E}$. That is, a trust relationship exists from a consumer towards another consumer or a provider. We do not consider trust relationships from providers towards consumers.

Let's consider the trust of an entity a in an entity b in the context of an action ψ given as the tuple $\langle (a, b) \in \mathbb{T}, \psi, l_{a,b,\psi} \rangle$. If $b \in \mathbb{C}$ then $\psi = \rho$, else if $b \in \mathbb{P}$ then $\psi \in \Psi_{sp}$. That is, consumers have trust among themselves in the context of preserving each other's privacy. Consumers have trust in web service providers in the context of providing web services with certain qualities.

Let $\mathbb{L} = \{0.0, 0.25, 0.5, 0.75, 1.0\}$. Let $l_{a,b} \in \mathbb{L}$. That is, trust can be quantified as one of the five values in $\mathbb{L}$. The five values may be interpreted as follows: 0.0 = "no trust", 0.25 = "low trust", 0.5 = "medium trust", 0.75 = "high trust", 1.0 = "absolute trust".

D. Reputation

Let X be a discrete random variable on the discrete sample space $\mathbb{L}$. A Probability Mass Function (PMF) gives the probability of the occurrence of each of the values of a discrete random variable. The probability mass function for X is given as follows:

$$f_X(x) = \begin{cases} Pr(X = 0.00), & x = 0.00 \\ Pr(X = 0.25), & x = 0.25 \\ Pr(X = 0.50), & x = 0.50 \\ Pr(X = 0.75), & x = 0.75 \\ Pr(X = 1.00), & x = 1.00 \end{cases} \quad (1)$$

Let S_t be the set of the source entities of agent t, that is, the set of entities who hold an opinion about agent t.

We define the reputation of a target provider entity t as the probability mass function $f_X(l_{a,t})$, where $a \in S_t$. The advantage of using PMF to represent reputation is that it gives a comprehensive overview of the opinion of consumers about the target provider. This is in contrast to reputation given as a single variable such as the sum or mean of the opinions, which does not portray a picture as comprehensive as the PMF does. An example of the PMF of a provider entity t with fairly good reputation is as follows:

$$f_X(l_{a,t}) = \begin{cases} 0, & l_{a,t} = 0.00 \\ 0, & l_{a,t} = 0.25 \\ 1/8, & l_{a,t} = 0.50 \\ 5/8, & l_{a,t} = 0.75 \\ 2/8, & l_{a,t} = 1.00 \end{cases} \quad (2)$$

Generally speaking, a probability mass function shows the probability of a random variable's value to fall in each specific range. In this paper we build the function by using opinions as estimates of the probabilities of a fuzzy variable (the reputation, which takes values in the [0,1] interval) to fall in a given range. This is consistent with mass-assignment based probability theory [14], and could in the future enable composing reputation with other random events having a PMF. This would also open the way to using conditional probability models to relate reputation to context.

E. Adversarial Model

We consider the standard semi-honest adversarial model. It is assumed that entities do not deviate from the specified protocol. In other words, they always execute the protocol according to the specifications. Entities also abstain from wiretapping and tampering of the communication channels. However, within these constraints, the dishonest entities passively attempt to learn the inputs of honest entities by using intermediate information received during the protocol and any other information that they can gain through legitimate means.

F. Privacy Preserving Reputation Protocol

We adopt the Ideal-Real approach [15], [16] to define a privacy preserving reputation protocol.

Intuitively, a multi-party protocol in the Ideal Model is a protocol that comprises of a Trusted Third Party (TTP) as a participant. The TTP receives all inputs in the protocol and then locally computes the output. On the other hand, a multi-party protocol in the Real Model is a protocol that does not rely on a TTP, and computes the output in a distributed manner.

The participants of an ideal privacy preserving reputation protocol are as follows: 1) a target provider entity t whose reputation is in question; 2) a querying consumer entity q who would like to learn the reputation $f_X(l_{a,t})$, where $a \in S_t$; 3) all entities in the set S_t, who are source consumer entities that have previously interacted with t and hold a trust relationship towards t; 4) a trusted third party, which is trusted by all of the source entities. The inputs of the protocol are as follows: each entity $a \in S_t$ provides the secret input $l_{a,t}$. The output of the protocol is as follows: the querying entity q learns the reputation of t.

The ideal protocol operates as follows: Each source entity $a \in S_t$ submits $l_{a,t}$ to the TTP. The TTP computes $f_X(l_{a,t})$ and delivers it to the querying entity q. The TTP is considered honest and fully trustworthy, therefore a dishonest entity cannot obtain any more information about the private inputs of honest participants other than what it can learn from the information that it holds beforehand and the output of the protocol.

A real protocol has the same participants, inputs, and outputs as the ideal protocol with the exception that a TTP is not available. The protocol is said to preserve the privacy of the participants if it can emulate the ideal protocol. Emulating the ideal protocol means that their is a high probability that the dishonest entities cannot obtain any more information about the private inputs of honest entities than they can learn in the ideal protocol.

The *security threshold* is a parameter that can be assigned a value in $[0, 1]$ according to the security needs of an application. A value of the *security threshold* closer to 1 indicates a stricter security requirement. We consider *high* probability as probability greater than or equal to the *security threshold*, and *low* probability as probability less than or equal to $1-$ *security threshold*. For example, if the *security threshold* is established at 0.90, then a probability greater than or equal to 0.90 would be considered *high* and a probability less than or equal to $1 - 0.90 = 0.10$ would be considered *low*.

G. Representation of $l_{a,b}$ by 1-of-K Encoded Vectors

Let $\overrightarrow{l}_{a,b}$ be a *1-of-K* encoded 5-dimensional vector. A *1-of-K* vector has the property that exactly one element has the value 1 and all other elements have the value 0. The position of 1 in $\overrightarrow{l}_{a,b}$ corresponds to the value of $l_{a,b}$. If $l_{a,b} = 0.0$ then $\overrightarrow{l}_{a,b} = \langle 1, 0, 0, 0, 0 \rangle$. If $l_{a,b} = 0.25$ then $\overrightarrow{l}_{a,b} = \langle 0, 1, 0, 0, 0 \rangle$ and so on up to the case where $\overrightarrow{l}_{a,b} = \langle 0, 0, 0, 0, 1 \rangle$ if $l_{a,b} = 1.0$. We denote the i^{th} element of a vector $\overrightarrow{v}$ as $\overrightarrow{v}[i]$.

III. The *WS-P-Rep* Protocol

In this section we present our *WS-P-Rep* protocol, which is a real privacy preserving reputation protocol for web service provider selection. The *WS-P-Rep* protocol is an adaptation of our more general k-Shares [17] reputation protocol. The two key differentiating characteristics of the *WS-P-Rep* protocol are as follows: 1) The framework for *WS-P-Rep* is specialized for the web services architecture. 2) The *WS-P-Rep* protocol treats reputation as a probability mass function. This is in contrast to k-Shares, which computes

reputation as the mean of the private inputs. The probability mass function gives a more comprehensive overview of the opinions of the consumers. The key steps of the *WS-P-Rep* protocol are outlined below.

1) **Initiate.** The protocol is initiated by a querying consumer entity q to determine the reputation of a target provider entity t. Entity q retrieves S_t, which is the set of source entities maintained by t. Entity q then sends the set S_t to each entity $a \in S_t$.
2) **Select Trustworthy Entities.** Each entity $a \in S_t$ selects k other entities from the set S_t. Let's refer to these entities selected by a as the set $U_a = \{u_{a,1} \dots u_{a,k}\}$. Entity a selects these entities such that: $(1 - l_{a,u_{a,1},\rho}) \times \ldots \times (1 - l_{a,u_{a,k},\rho})$ is low. That is, the probability that all of the k selected entities are dishonest is low.
3) **Prepare Shares.** Entity a then prepares $k+1$ shares of its secret vector $\overrightarrow{l}_{a,t}$. The shares are denoted as: $\overrightarrow{x}_{a,1} \dots \overrightarrow{x}_{a,k+1}$. They are prepared as follows: The first k shares are 5-dimensional vectors with elements that are random real numbers uniformly distributed over a large interval. The $k + 1^{th}$ share is a 5-dimensional vector whose elements are selected such that: $\sum_{i=1}^{k+1} \overrightarrow{x}_{a,i} = \overrightarrow{l}_{a,t}$. That is, the vector sum of the $k + 1$ shares is equal to the secret vector $\overrightarrow{l}_{a,t}$. Since each of the $k + 1$ shares is composed of numbers uniformly distributed over a large interval, no information about the secret is revealed unless all of the shares are known.
4) **Send Shares.** Each entity $a \in S_t$ sends the set $U_a = \{u_{a,1} \dots u_{a,k}\}$ to the querying entity q. Entity a also sends each share $\overrightarrow{x}_{a,i}$ to entity $u_{a,i}$, where $i \in \{1 \dots k\}$.
5) **Receive Shares.** Entity q receives U_a from each entity $a \in S_t$. Then for each entity $a \in S_t$, entity q: 1) compiles the list of entities from whom a should expect to receive shares, and 2) sends this list to entity a. Entity a then proceeds to receive shares from the entities on the list until all shares are received.
6) **Compute Sums.** Entity a computes $\overrightarrow{\sigma_a}$, which is the vector sum of all the shares received and its own final share $\overrightarrow{x}_{a,k+1}$. Entity a then sends the sum $\overrightarrow{\sigma_a}$ to q.
7) **Compute Reputation.** Entity q receives the sum $\overrightarrow{\sigma_a}$ from each entity $a \in S_t$. Entity q computes $\overrightarrow{\gamma} = \sum_{a \in S_t} \overrightarrow{\sigma_a}$. Entity q then computes the reputation of entity t as the probability mass function $f_X(l_{a,t})$.

$$f_X(l_{a,t}) = \begin{cases} \overrightarrow{\gamma}[1]/n, & x = 0.00 \\ \overrightarrow{\gamma}[2]/n, & x = 0.25 \\ \overrightarrow{\gamma}[3]/n, & x = 0.50 \\ \overrightarrow{\gamma}[4]/n, & x = 0.75 \\ \overrightarrow{\gamma}[5]/n, & x = 1.00 \end{cases} \quad (3)$$

A third difference in the *WS-P-Rep* protocol is that each agent a is required to select exactly k agents. This is in contrast to the k-Shares protocol, in which each agent a selects the minimum number of trustworthy agents upto k such that the security threshold is satisfied. Selecting only trustworthy agents reveals to the adversary that agent a has a high degree of trust in them. However, if exactly k agents are selected, as is the case in the *WS-P-Rep* protocol, then agent a can also include some random agents in the set of selected agents. This approach limits the adversary from ascertaining whether a given agent in a's selected set is an agent trusted by a or an agent that has been randomly selected.

A. Protocol Specification

The protocol is specified in Figure 1. $[-Y, Y]$ is an interval of real numbers. The function $random(-Y, Y)$ returns a random number on the interval $[-Y, Y]$. The function $set\_of\_trustworthy(a, S)$ returns a set of entities $U_a = \{u_{a,1} \dots u_{a,k}\}$, where $U_a \subseteq S$, and $k \ll n$ is a constant. The set U_a is selected such that: $(1 - l_{a,u_{a,1},\rho}) \times \ldots \times (1 - l_{a,u_{a,k},\rho})$ is low. In [17] we conducted experiments on a real web of trust (Advogato.org), which indicated that a high majority of entities are able to find enough trustworthy fellow entities under the constraint $k \ll n$ such that $(1 - l_{a,u_{a,1},\rho}) \times \ldots \times (1 - l_{a,u_{a,k},\rho})$ is low for them. The consequence of keeping $k \ll n$ is a protocol that requires $O(n)$ messages to be exchanged. This is in contrast to comparable reputation protocols [18], [19] where k is implicitly equal to n and as a consequence the number messages exchanged is $O(n^2)$.

B. Security Analysis

1) Correctness: Each entity $a \in S_t$ prepares the shares $\overrightarrow{x}_{a,1} \dots \overrightarrow{x}_{a,k+1}$ of its secret input $\overrightarrow{l}_{a,t}$, such that: $\sum_{j=1}^{k+1} \overrightarrow{x}_{a,j} = \overrightarrow{l}_{a,t}$. The vector sum of the private inputs of all entities in $S_t = \{a_1 \dots a_n\}$ is given as: $\sum_{i=1}^{n} \overrightarrow{l}_{a_i,t}$. The sum of the private inputs of all entities in S_t can be stated as: $\sum_{i=1}^{n} (\sum_{j=1}^{k+1} \overrightarrow{x}_{a_i,j})$. That is, the sum of all shares of all entities.

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

Protocol: WS-P-Rep

Participants: Web Service Entities: q, t, $S_t \equiv S_{t,\psi} = \{a_1 \dots a_n\}$. Entities 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 consumer entity $a \in S_{t,\psi}$ has a private input $l_{a,t}$.

Output: The querying consumer entity q learns $f_X(l_{a,t})$, the reputation of the target provider entity t in context ψ, where $a \in S_{t,\psi}$.

Setup: Each provider entity $t \in \mathbb{P}$ maintains $S_t \equiv S_{t,\psi}$, the set of its source consumer entities in context ψ.

Events and Associated Actions (for a Consumer Entity a):

need arises to determine $f_X(l_{a,t})$

▷ initiate query
1 send tuple (REQUEST_FOR_SOURCES, ψ) to t
2 receive tuple (SOURCES, ψ, S_t) from t
3 **for** each entity $v \in S_t$
4 **do** $J_v \leftarrow \phi$
5 $S'_t \leftarrow S_t$
6 $\overrightarrow{\gamma} \leftarrow \langle 0, 0, 0, 0, 0 \rangle$
7 $q \leftarrow a$
8 $s \leftarrow$ timestamp()
9 send tuple (PREP, q, t, s, S_t) to each entity $v \in S_t$

tuple (REQUEST_FOR_SOURCES, ψ) **received from entity** q
1 send tuple (SOURCES, ψ, S_a) to q

tuple (PREP, q, t, s, S_t) **received from entity** q
1 $I \leftarrow \phi$
2 $J \leftarrow \phi$
3 $\overrightarrow{\sigma}_a \leftarrow \langle 0, 0, 0, 0, 0 \rangle$
4 $U_a \leftarrow$ set_of_trustworthy(a, $S_t - a$)
5 **for** $i \leftarrow 1$ **to** k
6 **do** $\overrightarrow{x}_{a,i} \leftarrow \langle \text{random}(-Y, Y), \text{random}(-Y, Y), \text{random}(-Y, Y), \text{random}(-Y, Y), \text{random}(-Y, Y) \rangle$
7 $\overrightarrow{x}_{a,k+1} \leftarrow \overrightarrow{l}_{a,t} - \sum_{i=1}^{k} \overrightarrow{x}_{a,i}$
8 send tuple (RECIPIENTS, q, t, s, U_a) to entity q
9 **for** each entity $u_{a,i} \in U_a = \{u_{a,1} \dots u_{a,k}\}$
10 **do** send tuple (SHARE, $q, t, s, \overrightarrow{x}_{a,i}$) to entity $u_{a,i}$

tuple (RECIPIENTS, q, t, s, U_v) **received from an entity** $v \in S_t$
1 **for** each entity $u \in U_v$
2 **do** $J_u \leftarrow J_u \cup v$
3 $S'_t \leftarrow S'_t - v$
4 **if** $S'_t = \phi$
5 **then** $S'_t \leftarrow S_t$
6 **for** each entity $w \in S_t$
7 **do** send tuple (SENDERS, q, t, s, J_w) to entity w

tuple (SHARE, $q, t, s, \overrightarrow{x}_v$) **received from an entity** $v \in S_t$
1 $I \leftarrow I \cup v$
2 $\overrightarrow{\sigma}_a \leftarrow \overrightarrow{\sigma}_a + \overrightarrow{x}_v$
3 **if** $I = J$
4 **then** $\overrightarrow{\sigma}_a \leftarrow \overrightarrow{\sigma}_a + \overrightarrow{x}_{a,k+1}$
5 send tuple (SUM, $q, t, s, \overrightarrow{\sigma}_a$) to entity q

tuple (SENDERS, q, t, s, J_a) **received from entity** q
1 $J \leftarrow J_a$
2 **if** $I = J$
3 **then** $\overrightarrow{\sigma}_a \leftarrow \overrightarrow{\sigma}_a + \overrightarrow{x}_{a,k+1}$
4 send tuple (SUM, $q, t, s, \overrightarrow{\sigma}_a$) to entity q

tuple (SUM, $q, t, s, \overrightarrow{\sigma_v}$) **received from an entity** $v \in S_t$
1 $S'_t \leftarrow S'_t - v$
2 $\overrightarrow{\gamma} \leftarrow \overrightarrow{\gamma} + \overrightarrow{\sigma}_v$
3 **if** $S'_t = \phi$
4 **then** compute $f_X(l_{a,t})$ (Equation 3)

Figure 1. Protocol: WS-P-Rep

The final value of $\overrightarrow{\gamma}$ in the protocol is: $\overrightarrow{\gamma} = \sum_{i=1}^{n} \overrightarrow{\sigma}_{a_i} = \sum_{i=1}^{n} (\sum_{j=1}^{k+1} \overrightarrow{x}_{a_i,j}) = \sum_{i=1}^{n} \overrightarrow{l}_{a_i t}$. Thus when q computes $f_X(l_{a,t})$, it is the correct reputation of entity t in context ψ (Equation 3).

2) Privacy: Let's consider an entity $a \in S_t$. entity a prepares the shares $\overrightarrow{x}_{a,1} \dots \overrightarrow{x}_{a,k+1}$ of its private input $\overrightarrow{l}_{a,t}$. The elements of the first k shares $\overrightarrow{x}_{a,1} \dots \overrightarrow{x}_{a,k}$ are random numbers uniformly distributed over a large interval $[-Y, Y]$. The final share, $\overrightarrow{x}_{a,k+1} = \overrightarrow{l}_{a,t} - \sum_{i=1}^{k} \overrightarrow{x}_{a,i}$, is also composed of numbers uniformly distributed over a large interval because they are functions of the elements of the first k shares which are random numbers. Thus, individually each of the shares does not reveal any information about the private input $\overrightarrow{l}_{a,t}$. Moreover, no information is learnt about $\overrightarrow{l}_{a,t}$ even if up to k shares are known, since their sum would be some random numbers uniformly distributed over a large interval. The only case in which information can be gained about $\overrightarrow{l}_{a,t}$ is if all $k+1$ shares are known. Then, $\overrightarrow{l}_{a,t} = \sum_{i=1}^{k+1} \overrightarrow{x}_{a,i}$.

We now analyze if the $k+1$ shares of an entity a can be learnt by a dishonest entity from the protocol.

Entity a sends each share $\overrightarrow{x}_{a,i}$ only to entity $u_{a,i}$, where $i \in \{1 \dots k\}$. Each $u_{a,i}$ then computes $\overrightarrow{\sigma}_{u_{a,i}}$, which is the sum of all shares that it receives and its own final share $\overrightarrow{x}_{u_{a,i},k+1}$. Even if entity a is the only entity to send entity $u_{a,i}$ a share, $\overrightarrow{\sigma}_{u_{a,i}} = \overrightarrow{x}_{a,i} + \overrightarrow{x}_{u_{a,i},k+1}$. That is, the sum of entity a's share and entity $u_{a,i}$'s final share. $\overrightarrow{\sigma}_{u_{a,i}}$ is composed of numbers uniformly distributed over a large interval. Thus, when entity $u_{a,i}$ sends this vector sum to entity q, it is impossible for q to distinguish the individual shares from the vector sum. Therefore, each share $\overrightarrow{x}_{a,i}$ that entity a sends to entity $u_{a,i}$ will only be known to entity $u_{a,i}$. Unless, entity $u_{a,i}$ is dishonest. The probability that entity $u_{a,i}$ is dishonest, that is, it will attempt to breach entity a's privacy is given as: $1 - l_{a,u_{a,i},\rho}$.

To learn the first k shares of entity a, all entities $u_{a,1} \dots u_{a,k}$ would have to be dishonest. The probability of this scenario is: $(1 - l_{a,u_{a,1},\rho}) \times \dots \times (1 - l_{a,u_{a,k},\rho})$.

Even in the above scenario, the adversary does not gain information about $\overrightarrow{l}_{a,t}$, without the knowledge of entity a's final share $\overrightarrow{x}_{a,k+1}$. However, entity a has to send $\overrightarrow{\sigma}_a = \overrightarrow{x}_{a,k+1} + \sum_{v \in J_a} \overrightarrow{x}_v$, and entity a has no control over the $\sum_{v \in J_a} \overrightarrow{x}_v$ part of the equation. Therefore, we assume that entity q learns the final share of entity a.

Thus the probability that the protocol will not preserve entity a's privacy can be stated as: $(1 - l_{a,u_{a,1},\rho}) \times \dots \times (1 - l_{a,u_{a,k},\rho})$. If we assume that the entities $u_{a,1} \dots u_{a,k}$ are selected such that this probability is low, then with high probability, the adversary learns no more information about $\overrightarrow{l}_{a,t}$ than it can learn in an ideal protocol with what it knows before the execution of the protocol and the outcome.

Privacy also derives by observing that every agent $a \in S_t$ uses an independent $(k+1, k+1)$ threshold secret sharing scheme [20]. Thus, the adversary cannot learn private information even if it collects upto k shares of any agent.

C. Complexity Analysis

The protocol requires $4n + kn + 2$ or $O(n)$ messages to be exchanged. In terms of bandwidth usage, the protocol requires transmission of $O(n^2)$ entity IDs and $O(n)$ real numbers. The protocol is more efficient in relation to comparable reputation protocols for decentralized environments such as the ones proposed by Pavlov et al. [18] and Gudes et al. [19], both of which require $O(n^2)$ messages to be exchanged. Their bandwidth usage is also higher as they require transmission of $O(n^2)$ entity IDs and $O(n^2)$ numbers.

IV. Conclusion

Our contribution in this article is a reputation protocol (WS-P-Rep) for web service provider selection that preserves the privacy of the consumers who provide feedback. The advantage of privacy is that consumers can provide truthful feedback without the fear of reprisal from the target entities. The protocol is decentralized as well as efficient. The protocol is an adaptation of our more general k-shares reputation protocol. However, the WS-P-Rep protocol is different from the k-shares protocol in two key aspects: 1) Its framework is specialized for the web services architecture. 2) It treats reputation as a probability mass function, which provides a more comprehensive overview of the opinions. This is in contrast to the k-shares protocol, which represents reputation as the mean of the opinions.

References

[1] Y. Wang and J. Vassileva, "Toward trust and reputation based web service selection: A survey," *Journal of Multi-agent and Grid Systems (MAGS) , Special Issue on "New tendencies on Web Services and Multi-agent Systems (WS-MAS)"*, 2007.

[2] A. Pandey and S. K. Jena, "Dynamic approach for web services selection," in *Proceedings of the International MultiConference of Engineers and Computer Scientists (IMECS 2009)*, Hong Kong, March 18 - 20 2009.

[3] Y. Badr, A. Abraham, F. Biennier, and C. Grosan, "Enhancing web service selection by user preferences of non-functional features," in *Proceedings of the 4th International Conference on Next Generation Web Services Practices (NWESP '08)*, 2008.

[4] T. Yu, Y. Zhang, and K.-J. Lin, "Efficient algorithms for web services selection with end-to-end qos constraints," *ACM Transactions on the Web (TWEB)*, vol. 1, no. 1, May 2007.

[5] K. Karta, "An investigation on personalized collaborative filtering for web service selection," The University of Western Australia, Tech. Rep., 2005.

[6] U. S. Manikrao and T. Prabhakar, "Dynamic selection of web services with recommendation system," in *Proceedings of the International Conference on Next Generation Web Services Practices (NWESP '05)*, 2005.

[7] E. M. Maximilien and M. P. Singh, "Multiagent system for dynamic web services selection," in *Proceedings of the 1st Workshop on Service-Oriented Computing and Agent-Based Engineering (SOCABE)*, 2005.

[8] Y. Liu, A. H. Ngu, and L. Z. Zeng, "Qos computation and policing in dynamic web service selection," in *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters (WWW Alt. '04)*, 2004.

[9] W. Han, "Integrating peer-to-peer into web services," Master's thesis, University of Saskatchewan, 2006.

[10] L.-H. Vu, M. Hauswirth, and K. Aberer, "Towards p2p-based semantic web service discovery with qos support," in *Proceedings of the Workshop on Business Processes and Services (BPS)*, Nancy, France, 2005.

[11] F. B. Kashani, C. C. Chen, and C. Shahabi, "Wspds: Web services peer-to-peer discovery service," in *Proceedings of the International Conference on Internet Computing*, 2004.

[12] P. Resnick and R. Zeckhauser, "Trust among strangers in internet transactions," *The Economics of the Internet and E-Commerce. Vol. 11 of Advances in Applied Microeconomics*, pp. 127–157, 2002.

[13] D. Gambetta, *Trust: Making and Breaking Cooperative Relatioins*. Department of Sociology, University of Oxford, 2000, ch. Can We Trust Trust?, pp. 213 – 237.

[14] J. F. Baldwin, J. Lawry, and T. P. Martin, "A mass assignment theory of the probability of fuzzy events," *Fuzzy Sets and Systems*, vol. 83, no. 3, pp. 353 – 367, November 1996.

[15] O. Goldreich, *The Foundations of Crypto. - Vol. 2*. Cambridge Univ. Press, 2004.

[16] R. Canetti, "Universally composable security: A new paradigm for cryptographic protocols," in *Proceedings of the 42nd Annual IEEE Symposium on the Foundations of Computer Science (FOCS'01)*, 2001.

[17] O. Hasan, L. Brunie, and E. Bertino, "k-shares: A privacy preserving reputation protocol for decentralized environments," in *Proceedings of the 25th IFIP International Information Security Conference (SEC 2010)*, Brisbane, Australia, September 20 - 23 2010.

[18] E. Pavlov, J. S. Rosenschein, and Z. Topol, "Supporting privacy in decentralized additive reputation systems," in *Proc. of the 2nd Intl. Conf. on Trust Management (iTrust 2004)*, 2004.

[19] E. Gudes, N. Gal-Oz, and A. Grubshtein, "Methods for computing trust and reputation while preserving privacy," in *Proceedings of the 23rd Annual IFIP WG 11.3 Working Conference on Data and Applications Security*, 2009.

[20] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, p. 612613, 1979.