A Privacy Preserving Distributed Reputation Mechanism

Emmanuelle Anceaume, Gilles Guette, Paul Lajoie Mazenc, Nicolas Prigent, Valérie Viet Triem Tong

Publications Internes de l'IRISA
ISSN : 2102-6327
PI 2000 – Décembre 2012

A Privacy Preserving Distributed Reputation Mechanism

E Anceaume* G. Guette** P. Lajoie-Mazenc*** N. Prigent**** V. Viet Triem Tong*****

Abstract: Reputation systems allow to estimate the trustworthiness of entities based on their past behavior. Electronic commerce, peer-to-peer routing and collaborative environments, just to cite a few, highly benefit from using reputation systems. To guarantee an accurate estimation, reputation systems typically rely on a central authority, on the identification and authentication of all the participants, or both. In this paper, we go a step further by presenting a distributed reputation mechanism which is robust against malicious behaviors and that preserves the privacy of its clients. Guaranteed error bounds on the estimation are provided.

Key-words: Distributed systems, Reputation, Privacy

Système de réputation distribué préservant la vie privée

Résumé : *Dans ce travail, nous considérons le cas d'un réseau dans lequel des fournisseurs de services fournissent des services aux autres membres du réseaux que nous appelons les clients. Nous proposons un système de réputation permettant aux clients de décider s'ils peuvent en toute confiance interagir avec un fournisseur de service. Ce système de réputation est distribué afin qu'il n'existe pas de point unique de défaillance sur lequel un attaquant pourrait focaliser ses efforts. De plus, pour des raisons de vie privée notre proposition garantit l'anonymat des clients : il est impossible de lier l'identité d'un agent avec les interactions qu'il a eu avec différents fournisseurs de service ni avec les différents témoignages qu'il a pu émettre.*

Mots clés : *Mécanisme de réputation, distribution, respect de la vie privée*

---

* CNRS UMR 6074 IRISA, emmanuelle.anceaume@irisa.fr, CIDRE
** Université Rennes 1 UMR 6074 IRISA, gilles.guette@irisa.fr, CIDRE
*** Université Rennes 1 UMR 6074 IRISA, plajoiem@gmail.com, CIDRE
**** Supélec, nicolas.prigent@supelec.fr, CIDRE
***** Supélec, valerie.viettriemtong@supelec.fr, CIDRE

1 Introduction

In large scale and dynamic networks such as the Internet, most interactions occur between unknown users. When users invest time or money in such interactions, this induces a severe risk. For instance, in e-commerce transactions, a buyer has no idea of the real state of the item to be sold. This item can be more damaged than advertised, second-hand instead of brand new, etc. Hence the need for users to determine to what extent an interaction with a given user is safe or not.

Digital reputation mechanisms have recently emerged as a promising approach to cope with the specificities of large scale and dynamic systems. Similarly to real world reputation, a digital reputation mechanism expresses a collective opinion about a target user based on aggregated feedback about his past behavior. The resulting *reputation score* is usually a mathematical object, *e.g.* a number or a percentage. It is used to help entities in deciding whether an interaction with a target user should be considered. Digital reputation mechanisms are thus a powerful tool to incite users to trustworthily behave. Indeed, a user who behaves correctly improves his reputation score, encouraging more users to interact with him. In contrast, misbehaving users have lower reputation scores, which makes it harder for them to interact with other users.

To be useful, a reputation mechanism must itself be accurate against adversarial behaviors. Indeed, a user may attack the mechanism to increase his own reputation score or to reduce the reputation of a competitor. A user may also free-ride the mechanism and estimate the reputation of other users without providing his own feedback. Solutions that aim at preventing such attacks have been proposed in the literature. They usually exploit information redundancy techniques [**?**], robust reputation score functions [**?**], or cooperation incentives [**?**]. From what has been said, it should be clear that reputation is beneficial in order to reduce the potential risk of communicating with almost or completely unknown entities. Unfortunately, the user privacy may easily be jeopardized by reputation mechanisms which is clearly a strong argument to compromise the use of such a mechanism. Indeed, by collecting and aggregating user feedback, or by simply interacting with someone, reputation systems can be easily manipulated in order to deduce user profiles. Quoting Steinbrecher [**?**], "these profiles may include all the contexts in which the user has been involved in (for instance people or services with whom or which that user has lately interacted, frequency of these interactions). Deducing user profiles may be of high interest and a promising target for numerous data collectors or worse for retaliation arguments, but in any case is clearly contradictory with the user right to privacy". Furthermore, by protecting the identity of contributors and by maintaining unlinkability of their actions, it should give contributors incentives to feed the reputation mechanism with honest feedback without fearing retaliation.

Thus preserving user privacy while computing robust reputation is a real and important issue that this paper addresses. Our proposition combines techniques and algorithms coming from both distributed systems and privacy research domains. Specifically, we propose to self-organize agents over a logical structured graph, and to exploit properties of these graphs to anonymously store interactions feedback. By relying on robust reputation scores functions we tolerate ballot stuffing, bad mouthing and repudiation attacks. Finally, we guarantee error bounds on the reputation estimation score.

The remaining of the paper is organized as follows. Section 2 presents existing works in reputation systems and privacy. Section 3 formally defines the terminology used and presents our objectives. Our proposition is detailed in Section 4. Accuracy and privacy-preserving properties are proven in Section 5. Finally, Section 6 concludes and presents future works.

2 Related Work

The eBay [**?**] e-commerce website implements one of the most well-known reputation mechanisms. A transaction involves three entities: a *service provider* (*i.e.* the seller), a *client* (*i.e.* the buyer), and eBay's servers. The client starts by requesting the service provider's reputation score from eBay's servers. If this score fits the client's requirements, both the client and the provider proceed with the transaction. Otherwise, the interaction ends. Once the buyer has received the item and the seller has received the money, both can rate each other (in eBay, a feedback is a mark in $\{-1, 0, +1\}$).

In the beta reputation system [?] proposed by Jøsang and Ismail, a provider's behavior is modeled by a beta probability density function (pdf):

$$\text{beta}(p|a,b) = \frac{\Gamma(a+1)\Gamma(b+1)}{\Gamma(a+b+2)} p^a (1-p)^b$$

where a represents the amount of positive feedback, b the amount of negative ones and Γ is the function which extends the factorial to complex numbers. An example of such a reputation for $a = 7$ and $b = 1$ is shown on Fig. 1. It describes the probabilities of behaving benevolently for a service provider. This model is improved in [?], where ratings are continuous and their influence decreases over time. Furthermore, a filtering method allowing to ignore unfair ratings is described. A taxonomy of reputation systems is presented by Marti and Garcia-Molina in [?], while Jøsang [?] presents a comprehensive survey of these systems.

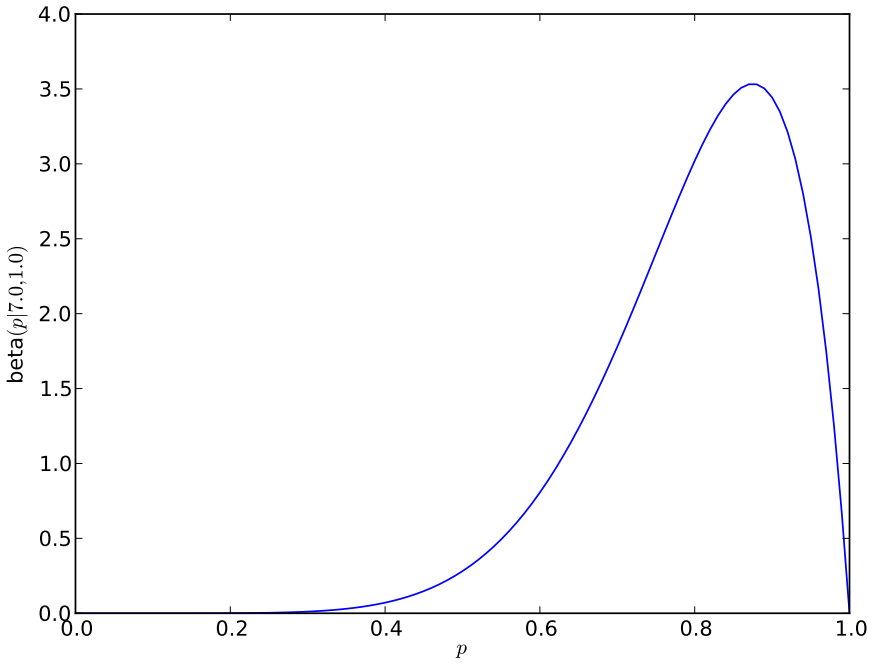

Figure 1: Example of beta reputation beta$(p|7,1)$

If no security mechanism is implemented, reputation systems are vulnerable to many attacks [?]. Among the most aggressive ones, bad-mouthing attacks consist in sending fallacious feedback against a service provider while ballot-stuffing attacks consist in creating many pieces of feedback to increase the reputation of a service provider. Finally, malicious clients may also repudiate a transaction. A lot of reputation mechanisms break down when raters collude [?]. Worse, the Sybil attack [?] allows an attacker to obtain the same power as a group of colluding attackers while being alone.

In addition to these attacks, the identification of users impedes their privacy. A solution to this problem is proposed by Androulaki *et al.* [?] through a central agent that stores the reputation of both clients and service providers. Pseudonyms, anonymous credentials and blind signatures allow to preserve the privacy of both clients and service providers. However, this solution presents the limitations of being centralized and only manipulating positive feedback. In contrast, we present a distributed reputation mechanism handling both positive and negative feedback and preserving clients' privacy.

3 Objectives

In this section, we start by presenting the terminology used in this work and then define the properties held by our reputation mechanism.

3.1 Terminology and definitions

A *feedback* is a data set allowing to evaluate a transaction. To obtain relevant reputation scores, multiple informations are needed, in particular the rating given by the client (p) on the service provider's (SP) behavior, the rating given by SP on the transaction, the time and the value of the transaction. To preserve their privacy, clients act under pseudonyms.

We propose a novel characterization of agents' behavior according to their honesty and correctness. Formally,

Definition 1 (Correctness)**.** *An agent is* correct *if he follows the protocol of the reputation system. He is* incorrect *otherwise.*

Note that correctness is a binary data, that is, during an interaction, an agent is either correct or incorrect.

Definition 2 (Honesty)**.** *Clients whose feedback reflect their judgement of the services offered by the providers are* honest*. Providers who would have judged good the service they provided during a transaction as if they were the client are* honest*. Otherwise, they are* dishonest*.*

This notion is subjective and continuous. The honesty of an agent is rated in $[0, 1]$. This notion can be compared with the English law notion of "reasonable person". An agent is said *benevolent* if he behaves honestly and correctly. He is *malicious* otherwise.

The two privacy properties our system guarantees are the following ones.

Definition 3 (Anonymity [**?**])**.** *An entity is* anonymous *if she is known only by pseudonyms,* i.e. *identifiers different from her real identity.*

Definition 4 (Unlinkability [**?**])**.** *Two different entities are* unlinkable *if an attacker is unable to determine whether they represent the same entity.*

3.2 Objectives

This work presents an *accurate* and *privacy-preserving distributed reputation system.* Formally, our reputation system is characterized by the following two properties.

Property 1 (Accuracy)**.** *The reputation score of a service provider* SP *calculated by a user eventually reflects his behavior with a known error bound. Formally, let t_h be the number of fallacious feedback emitted by malicious clients, ε_h and ε_c two small constants, P_h (resp. P_c) be the probability that* SP *is honest (resp. correct), and S_h (resp. S_c) be* SP*'s reputation score as computed by a client, then*

$$\forall t_h, \varepsilon_h, \varepsilon_c, \exists t_0 \mid \forall t > t_0 \Rightarrow \begin{cases} |P_h - S_h(t_h, t)| < \varepsilon_h \\ |P_c - S_c(t_h, t)| < \varepsilon_c \end{cases} \quad (1)$$

Property 2 (Privacy-preserving interactions)**.** *The reputation system preserves the privacy of its clients, that is,*

- *clients are anonymous;*
- *an identity and a pseudonym are unlinkable;*
- *two pseudonyms are unlinkable.*

Note that this work concentrates on clients' privacy. We left for future work the privacy of service providers.

4 Proposition

We now detail the solution we propose to design an accurate and privacy-preserving reputation mechanism. First, we present how agents self-organize in the network according to their identifiers. We then detail the notion of *mailboxes* and how these mailboxes are used to store agents' feedback through an interaction protocol between clients and service providers. Finally we explain how the reputation score is computed.

4.1 Self-organization of agents

To deal with large scale and dynamic systems, the reputation mechanism orchestrates the service providers into an overlay network. An overlay network is a self-organized virtual network that allows nodes to communicate easily by using transparently the underlying network, *e.g.* the IP network service. The algorithm used by nodes to choose their neighbors and to route their messages defines the overlay topology. This topology is built according to structured graphs (*e.g.* tree, torus, or hypercube). Most structured overlays are based on Distributed Hash Tables [**?**, **?**, **?**]. Generally, a unique random identifier from an m-bit identifier space is assigned to each node. Identifiers are derived by using some standard cryptographic one-way hash function on the nodes' network address. The value of m ($m = 128$ for the standard MD5 function for instance) is large enough to make the probability of identifiers' collision negligible. Nodes self-organize within the graph according to a distance function $\mathcal{D}$ based on nodes' IDs (*e.g.* two nodes are neighbors if their IDs share some common prefix) and possibly other criteria such as geographical distance. In our case, service providers are randomly organized on a ring that can host up to 2^m agents. Fig. 2 illustrates this overlay with $m = 4$. However, note that the principles of our reputation mechanism can be applied to any other overlay as long as this overlay is structured. In most of overlays, localization is efficient, *i.e.* is done in a poly-logarithmic number of hops in the size of the system.

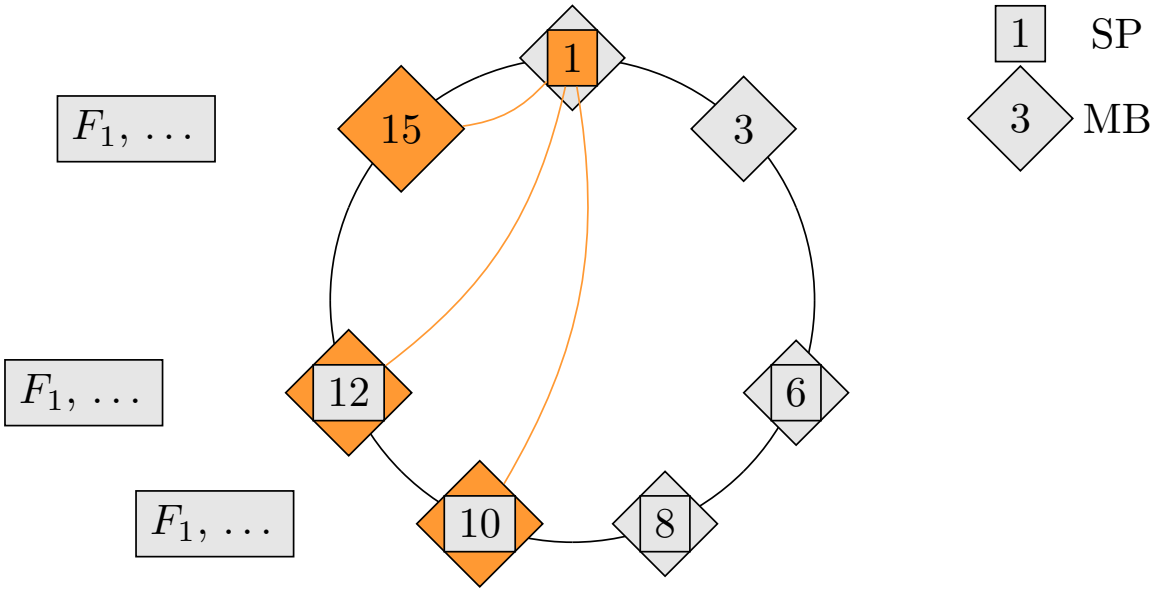

Figure 2: Architecture of the network

To preserve their anonymity and the unlinkability of their actions, clients interact only through pseudonyms and not with their identifier. Clients generate their own pseudonyms randomly as well as a set of public and private key. A certificate authority CA is used to certify the identifiers of each pseudonym and service provider, and allows any agent to authenticate another one by signing each agent's public key. In contrast to service providers who have a unique identifier, clients can generate as many pseudonyms as they wish in order to preserve their privacy. To discourage Sybil attacks, each certification requires a fee. The anonymity-preserving electronic currency Bitcoin [**?**] may be used for that purpose. Once a client has certified his pseudonyms $p_1, \ldots, p_\ell$, this client can start interacting with service providers. In the following, clients' pseudonyms are noted with lowercase letters, *e.g.* p, and service providers with uppercase letters, *e.g.* SP.

4.2 Mailboxes

To guarantee the accuracy of our mechanism despite the dynamicity and/or misbehaviors of agents, feedback about service providers' transactions are replicated in the network. Specifically, each service provider SP is associated with n agents that sit in the overlay. These agents, named *mailboxes*, are the n closest nodes to SP according to

the distance function $\mathcal{D}$ used in the overlay. In Fig. 2, the mailboxes are the predecessors of node 1, *i.e.* nodes 15, 12 and 10. The mailboxes are the recipients of transactions feedback, *e.g.* F_1 for the service provider 1. Note that service providers may act as the mailboxes of other providers. Hence, when a client p wishes to determine whether interacting with a given service provider SP is safe or not, p contacts SP's mailboxes and fetches the feedback they store regarding SP past behavior. To avoid ballot-stuffing attacks, a mailbox stores only the most recent feedback from a given client concerning a given provider. To cope with feedback manipulation by malicious mailboxes, p only keeps the feedback value that is the same in a quorum of mailboxes. It is well-known that assuming that no more than $n/3$ mailboxes behave maliciously guarantees integrity of the feedback [**?**]. Once collected, p computes SP's score as detailed in Section 4.4.

4.3 Interaction protocol

The interaction protocol between client p and service provider SP, whose mailboxes are noted $\mathrm{MB}_j, j \in \{1, \ldots, n\}$, is detailed in Fig. 3. This protocol is made of three steps.

The first one allows p to compute the reputation score of SP. To retrieve the feedback about SP, p contacts SP's mailboxes. Communications between p and $\mathrm{MB}_j, j \in \{1, \ldots, n\}$, should be protected in authenticity and confidentiality. For that purpose, the TLS protocol [**?**] is used.

Once secure communications between p and MB_j are established, p requests the feedback on SP to compute SP's reputation score. An accurate reputation score function is presented in Section 4.4. According to the value of the score, p decides whether he will engage in a transaction with SP. If so, p establishes a tunnel with SP. Client p then computes a transaction identifier $\tau = \mathrm{hash}(p \| \mathrm{SP} \| \mathrm{timestamp})$, '$\|$' being the concatenation operator. Both p and SP sign this identifier and communicate the identifier and the signature to all the MB_j. This ensures the commitment of both p and SP to the transaction. When at least $2n/3$ mailboxes have acknowledged this commitment, p and SP can engage in their transaction.

Once the transaction is finished, both p and SP send their ratings about it to SP mailboxes, that acknowledge it. Recall that a rating is made of two parts, one evaluating the correctness of the partner (*i.e.* p's rating $\rho^p_{\mathrm{SP},c}$ and SP's rating $\rho^{\mathrm{SP}}_{p,c}$), and the other one evaluating the honesty of the partner (*i.e.* p's rating $\rho^p_{\mathrm{SP},h}$ and SP's rating $\rho^{\mathrm{SP}}_{p,h}$). To prevent repudiation attacks, if either p or SP do not emit a feedback after a given timeout, the mailboxes provide a default rating. Namely, the client default rating is the maximum positive rating, while the provider's one is the maximum negative one. The feedback is then made available to everyone. We define a valid feedback as follows:

Definition 5 (Validity of a feedback)**.** *A feedback stored by the mailbox* MB_j *is* valid *if*

- *the feedback is issued from an actual transaction, and*
- *if both ratings have been received by* MB_j*, then* MB_j *did not manipulate. Otherwise,* MB_j *has provided default ratings for the missing ones*

Mailboxes synchronize feedback to maintain the same ratings everywhere. Given [**?**], we know that synchronization succeeds as long as no more than $n/3$ mailboxes behave maliciously.

4.4 Reputation calculation

Our proposition for the reputation calculation relies on [**?**] (see Section 2). As explained in Section 3.1, a feedback contains multiple pieces of information: the ratings of both the client and the provider, the age of the transaction and its value. In the following, we note $\rho^p_{\mathrm{SP},b}$ or $\rho^{\mathrm{SP}}_{p,b}$ a rating concerning the honesty or the correctness. In [**?**], a rating consists of two ratings. One is the positive rating while the second one is the negative rating. A rating ρ is divided as follows: $\rho = \begin{bmatrix}\rho_{+,b} & \rho_{-,b}\end{bmatrix}, \rho_{+,b}, \rho_{-,b} \in [0,1] \mid \rho_{+,b} + \rho_{-,b} = 1$. The parameters of the beta function are the sum of the positive ratings and the sum of the negative ones. Fig. 4 presents the general scheme of our method.

The first step is the modulation. Its inputs are the ratings of the provider about the client, $\rho^p_{\mathrm{SP},b}$, and of the client about the provider, $\rho^{\mathrm{SP}}_{p,b}$. The parameter is the correlation function f_{corr}. The objective of the modulation is to make the provider give an accurate rating of the transaction, and to punish her if she does not. Hence, the function has to add a bonus if the difference between the two ratings is low and a malus if it is high. For that purpose, the piecewise linear function which adds m_+ if $\rho^{\mathrm{SP}}_{p,b} = \rho^p_{\mathrm{SP},b}$, zero if $\left|\rho^p_{\mathrm{SP},b} - \rho^{\mathrm{SP}}_{p,b}\right| = \ell$ and m_- if $\left|\rho^p_{\mathrm{SP},b} - \rho^{\mathrm{SP}}_{p,b}\right| = 1$ can be used. Fig. 5 shows the modulated rating as a function of the client's and provider's ratings for $m_+ = 0.05$, $\ell = 0.1$ and $m_- = -0.6$. This figure shows that when a service provider behaves maliciously, he should better give an appropriate rating. For instance, suppose that a client gives a rating $\rho^{\mathrm{SP}}_{p,b} = 0.4$. Then SP would have a modulated rating of only 0.37 by lying and giving a rating $\rho^p_{\mathrm{SP},b} = 1.0$, while SP would have a modulated rating of 0.5 by telling the truth.

To give greater importance to high-valued transactions (for example based on their price in an e-commerce system), the ratings are weighted. If $\tilde{\rho}$ is the modulated note based on the ratings $\rho^p_{\mathrm{SP},b}$ and $\rho^{\mathrm{SP}}_{p,b}$ and w is the weight of the transaction, the weighted rating is $\hat{\rho} = \tilde{\rho} \times w$.[1]

The third step is to take the age of the transaction into account. Whitby *et al.* propose to use this aging function $f : t \mapsto \lambda^t,\ \lambda \in]0,1]$ [**?**]. If t_ρ is the elapsed time since a feedback was emitted, the aged feedback is $\bar{\rho} = \hat{\rho} \times \lambda^{t_\rho}$.[1]A value of λ close to 0 gives more emphasis to recent ratings.

Finally, to prevent bad-mouthing attacks, unfair ratings are filtered. Whitby *et al.* propose a method to filter "unfair ratings" [**?**]. Their algorithm regroups a client's feedback to compute a *local score* with these feedback. The local score is then compared with the *global score*, *i.e.* the score using all ratings. If the 5th percentile[2] of the local score is greater than the mean of the global one, or if the 95th percentile is lower than the mean, the client's feedback are filtered. This method is very accurate. However, it relies on the identification of clients, which is impossible in our context. To face this feature, we extend this method by multiplying each feedback by a filtering factor f_F. The simulation shown in Fig. 6 uses a filtering factor $f_F = 5$ to efficiently filter fallacious feedbacks when 15% of clients are malicious.

5 Accuracy of the System and Privacy of its Users

5.1 System accuracy

To prove that our system achieves accuracy as defined in Prop. 1, we proceed in two steps. First, we prove that a client obtains valid pieces of feedback from to the mailboxes. By construction, a benevolent mailbox stores a feedback if and only if this feedback corresponds to an actual transaction, *i.e.* to a transaction committed by a service provider and a client. Therefore, after the timeout, the feedback is valid as defined in Def. 5. By hypothesis, at least $2n/3$ mailboxes are benevolent, thus if they store a feedback, they give it to any client who asks for it.

We now prove that the computed reputation score is close enough to the behavior of SP. In the following, we focus on the precision error concerning the honesty of SP (the same argument holds for SP's correctness). Given the precision error ε_h and a number of bad-mouthing pieces of feedback t_h, there exists a threshold number of feedback t_0 such as the difference between the computed scores S_h and the behavior P_h is less than the precision error:

$$\forall t_h, \varepsilon_h, \exists t_0 \mid \forall t > t_0 \Rightarrow |P_h - S_h(t_h, t)| < \varepsilon_h \tag{2}$$

To prove relation (2), we consider a scenario where a service provider interacted with t different pseudonyms, one per time unit. The only difference between SP's behavior and the feedback about SP occurs when clients bad-mouth about the transaction. The worst-case scenario hence happens when SP behaved honestly while the pseudonyms bad-mouthed during the most recent rounds. The precision error is therefore:

[1]Recall that a vector $\begin{bmatrix}\rho_+ & \rho_-\end{bmatrix}$ times a scalar k is $\begin{bmatrix}\rho_+ \times k & \rho_- \times k\end{bmatrix}$

[2]The p-th percentile of a random variable distribution X is the smallest x such as $P(X \leq x) \geq p$.

$$|P_h - S_h| = \underbrace{\overbrace{\left|\frac{\sum_{k=1}^{t_h} \lambda^k}{\sum_{k=1}^{t} \lambda^k}\right|}^{\text{bad-mouthing ratings}}}_{\text{sum of all the ratings}} = \left|\frac{1-\lambda^{t_h}}{1-\lambda^t}\right|$$

To measure this accuracy, we model the behavior of a service provider with a probabilistic automaton and perform experiments in that model. Suppose that the provider behaves honestly with probability $P_h = 0.95$. When he is honest (in 95% of the transactions), his behavior is worth a rating uniformly distributed in the interval $I_h = [0.8, 1]$. Otherwise, it is worth a rating in $I_{\neg h} = [0.4, 0.8]$. When a client who interacts with SP is not malicious, she simply adds a bias chosen with a normal law of parameters $\mu = 0$ and $\sigma = 0.03$ to the previous rating. Otherwise, she rates the provider with a uniformly random rating in $[0, 0.1]$.

In our experiments, there are 1000 clients; 15% of them are malicious and bad-mouth about the transaction. The simulation is divided in 100 rounds. During a round, 10 clients are chosen randomly and make a transaction of value 1 with the provider. The parameters used for the reputation calculation are as follows: $m_+ = 0.05$, $\ell = 0.1$, $m_- = -0.6$, $\lambda = 0.9$ and $f_F = 5$.

Fig. 6 compares the evolution of SP's behavior and of her score as computed by a client. Three interesting phenomenon can be observed. During the five first rounds, some clients bad-mouth about the provider, which explains why the score decreases. During rounds 15 to 25, SP behaves dishonestly but her score decreases more than necessary. Indeed, as SP's score decreases, the initial bad-mouthing are no longer filtered. The last phenomenon occurs between rounds 60 and 70. Namely, the score computed by a client stays stable while the providers' behavior quickly decreases. This is due to the fact that the filtering algorithm filters the last transaction where SP behaved dishonestly. After a few rounds, the number of feedback that agree about the provider's bad behavior increases. At the same time, old feedback gets obsolete and the reputation score of SP matches the real behavior of SP. From this moment on, the reputation score matches accurately the behavior of SP.

5.2 Privacy concerns

According to Prop. 2, we consider a client's privacy to be preserved if three conditions are verified: anonymity, unlinkability between a pseudonym and an identity and unlinkability between two pseudonyms.

In our proposition, a client never uses her identity to communicate with other users, but rather use pseudonyms she generated herself. Thus, a client's *anonymity* is ensured, as well as the *unlinkability* between a pseudonym and an identity.

The last property requires that an attacker cannot know whether two different pseudonyms belong to the same identity. Many inference attacks exist, for which an example can be found in [?]. Inference attacks are based on analyses which reveal details allowing one to identify anonymized entities. The impact of such attacks on our system has not been studied yet, but we suppose they might be led by comparing feedback's information such as temporal habits or a client's behavior: an optimistic client will over rate transactions while a grumpy one will under rate them. These inference attacks are currently out of the scope of our proposal.

6 Conclusion and Future Work

In this article, we have proposed a reputation mechanism addressing the three main issues encountered in reputation systems: *distribution*, *accuracy* of the reputation score and *privacy* of the clients. This is achieved thanks to the introduction of mailbox agents and to the protocol we proposed. The system also protects from generic attacks on reputation mechanism such as Sybil attacks, bad-mouthing, or ballot-stuffing.

In future works, we will extend the work of Michiardi *et al.* [**?**] to design incentives for providing accurate ratings about transactions, despite anonymity. We will also study the impact of inference attacks on the privacy of the users of our system and will evaluate if signatures of reputation as proposed by Bethencourt *et al.* [**?**] can be used to anonymize service providers.

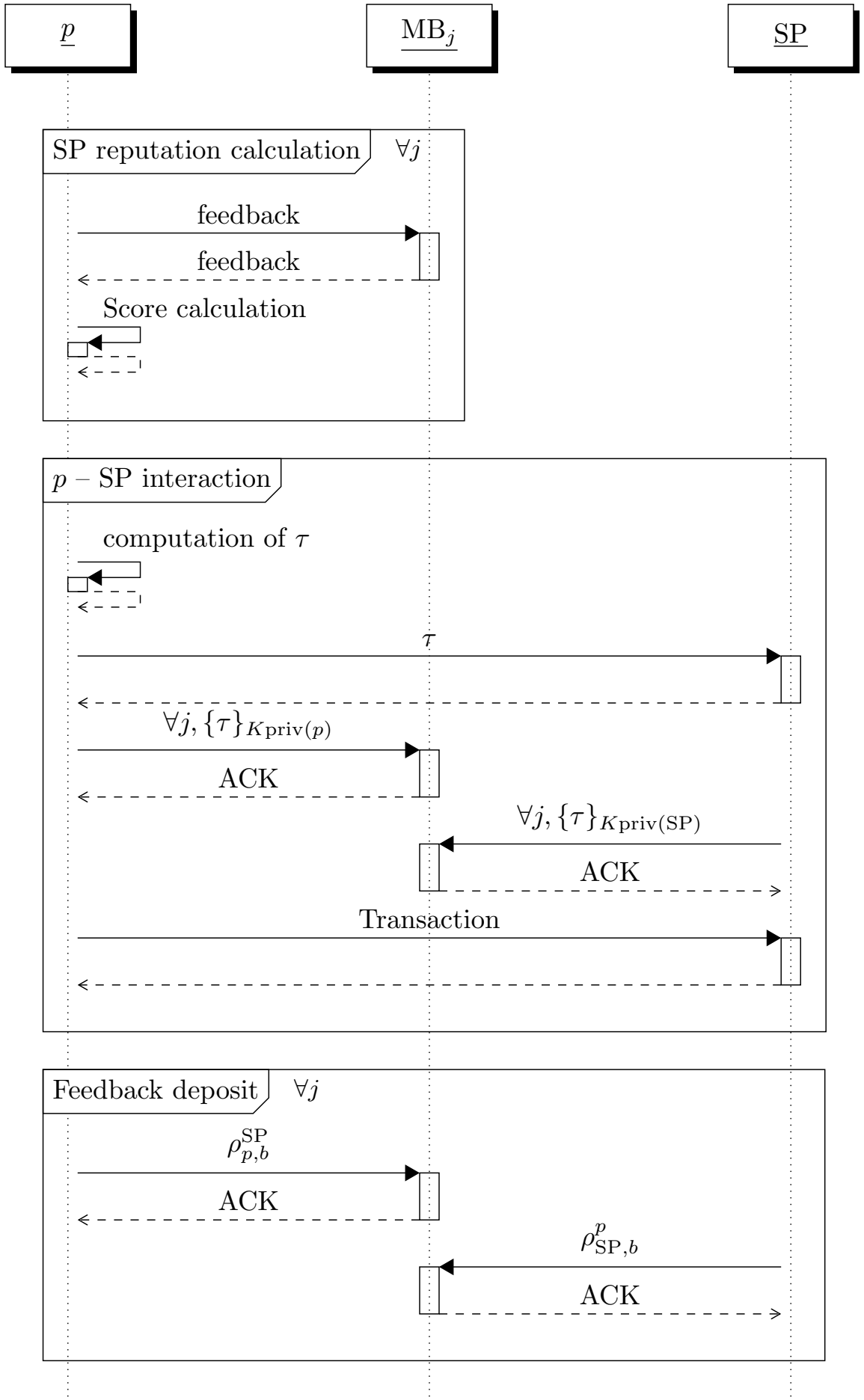

Figure 3: Interaction protocol between a client p, a service provider SP and her mailboxes MB_j, $1 \leq j \leq n$

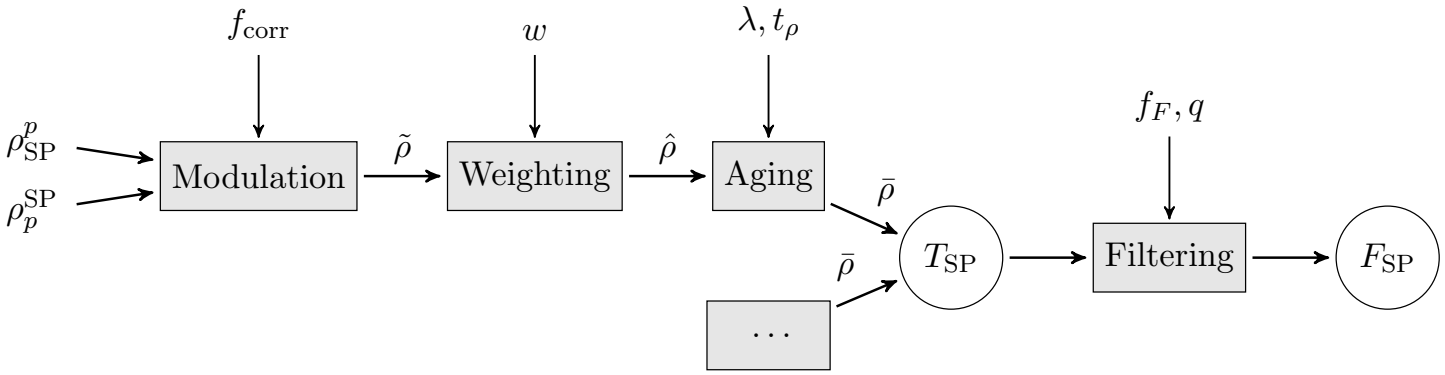

Figure 4: Reputation calculation steps

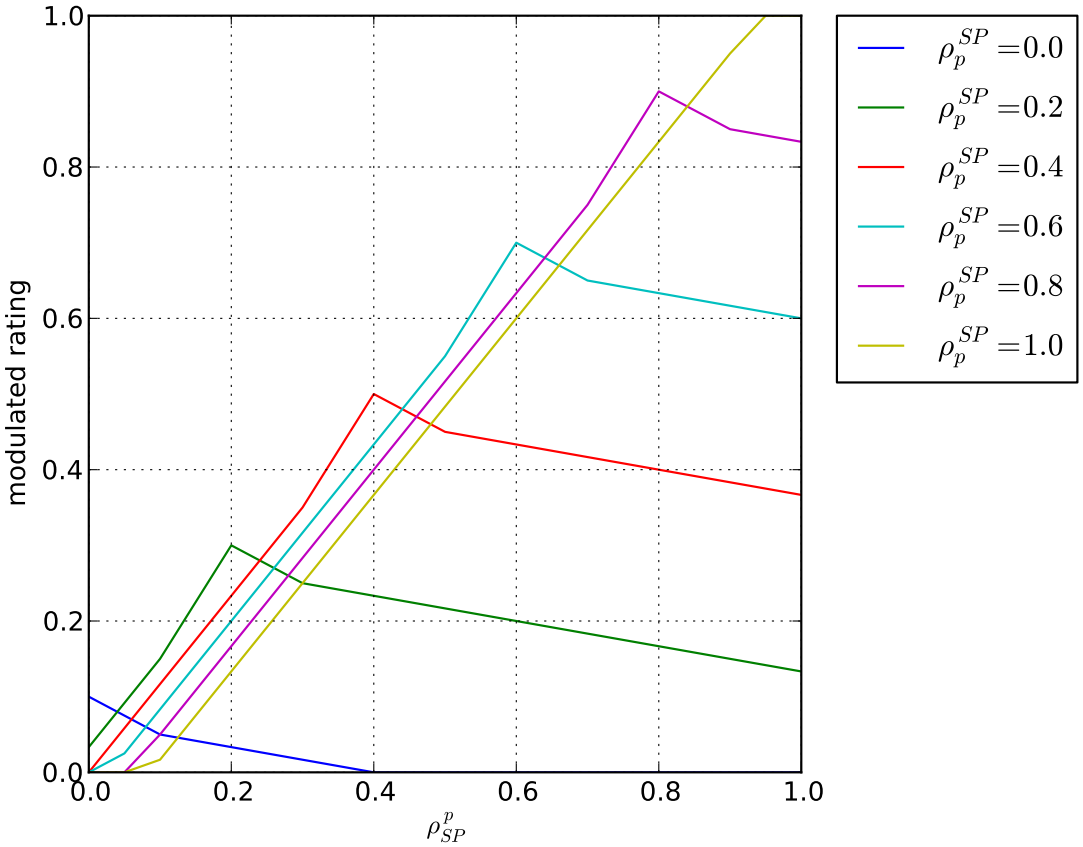

Figure 5: Correlation function

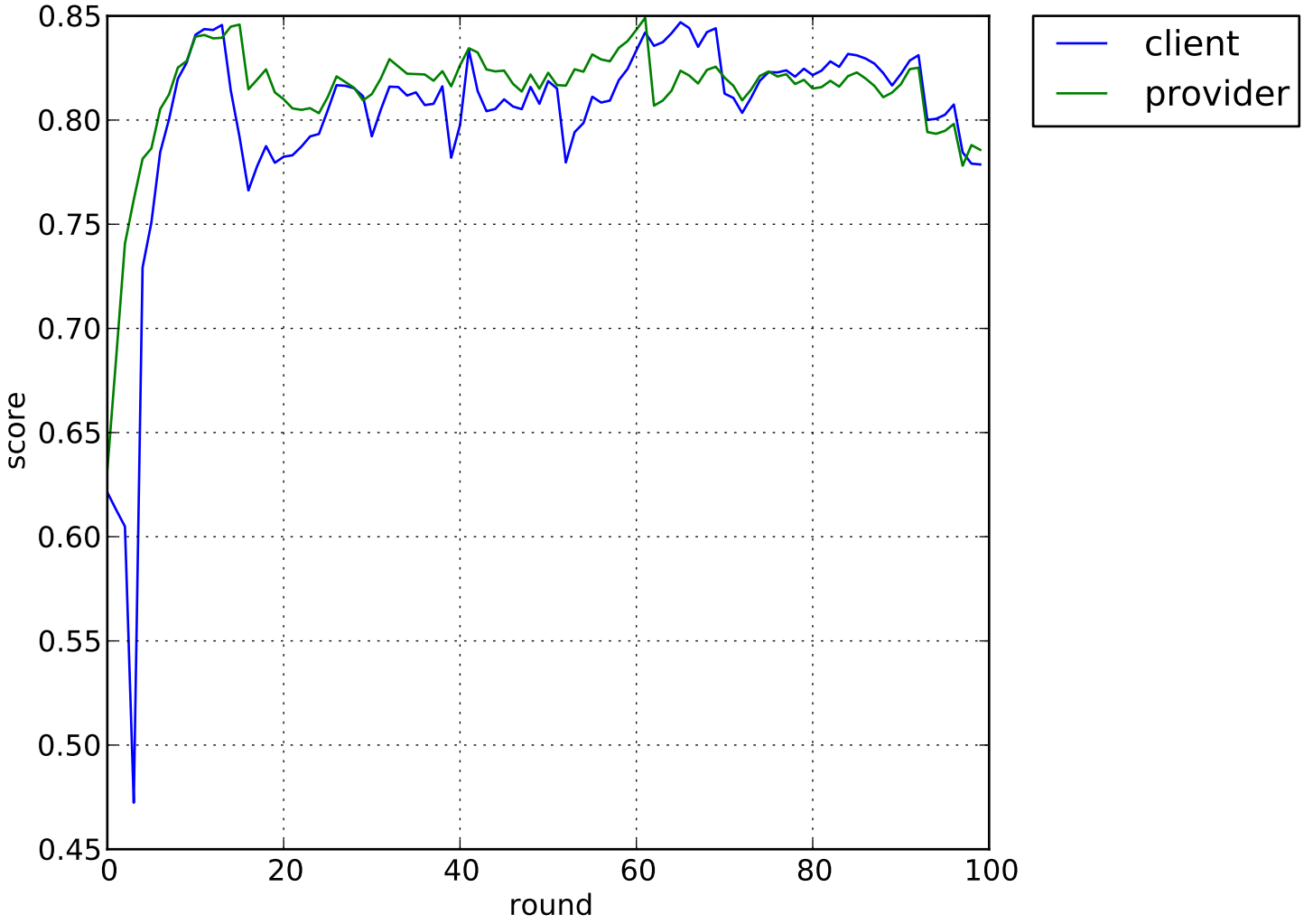

Figure 6: Evolution of a service provider's score