Privacy-Preserving and Co-Utile Distributed Social Credit

Josep Domingo-Ferrer

UNESCO Chair in Data Privacy
Department of Computer Science and Mathematics
Universitat Rovira i Virgili
Av. Països Catalans 26
E-43007 Tarragona, Catalonia
josep.domingo@urv.cat

Abstract. Reputation is a powerful incentive for agents to abide by the prescribed rules of an interaction. In computer science, reputation can be phrased as being an artificial incentive that can turn into self-enforcing protocols that would not be such otherwise. Quite recently, China has announced a national reputation system that will be launched in the future under the name of social credit system. However, to be generalizable without damaging the privacy of citizens/agents, a reputation system must be decentralized and privacy-preserving. We present a peer-to-peer fully distributed reputation protocol in which the anonymity of both the scoring and the scored agents is maintained. At the same time, the reputation protocol itself is co-utile, that is, the rational option for all agents is to honestly fulfill their part in the protocol.

Keywords: Protocols, Reputation, P2P, Self-enforcement, Co-utility, Privacy.

1 Introduction

Ensuring that agents will honestly follow their roles in an interaction has always been a thorny issue. In the presence of a common legal framework, the agents may have an incentive not to deviate from the legally established procedures. However, in an open environment such as the information society, common legal frameworks are often lacking. In this case, the prescribed rules of interaction, called a *protocol* in the computer science jargon, must be designed to that they deter deviations and are thus *self-enforcing*.

Reputation is a powerful incentive for agents to adhere to the prescribed interaction rules, both in the information society and in society at large. The Chinese government has realized this and they have recently announced a national reputation system called Social Credit System [1]. However, unless properly implemented, such a system has the potential of becoming a mass surveillance system. Technical details of the Chinese reputation system are not yet known, although it may be partly inspired on Alibaba's Sesame Credit [12]. Anyway, since no standard social credit system exists yet, it makes a lot of sense to investigate what properties a generalized reputation system should satisfy in order to be socially acceptable.

Privacy preservation and decentralization, or even better distributedness, stand out as obvious desiderata, and they are intertwined:

- Privacy should be preserved as much as possible for those agents whose reputation is computed: the legitimate interest of society is to know how well a certain agent has performed in the past, but this does not require knowing the detail of all the transactions the agent has been involved in. Whereas in a centralized reputation system, there is no way around the central authority learning the behavior in all transactions, a distributed system may be more privacy-preserving. After all, reputation is an aggregate metric of behavior, not a history of the behavior of an agent in each and every past transaction.
- Privacy should also be preserved for those agents participating in the computation of the reputation of other agents. Whereas in a centralized system it does not make sense to request privacy for the central authority computing reputations, in a decentralized system reputation may be computed by peers. Unless the privacy of those peers is protected, they might be subject to bribery, extortion or retaliation aimed at altering the reputations they compute.

1.1 Contribution and plan of this paper

In this paper, we describe a peer-to-peer reputation system that preserves privacy both for the agents involved in computing the reputations and for the agents whose reputation is computed. This makes it a good candidate to implement a generalized social credit system.

Section 2 recalls the basics of co-utility, which is a property related to a protocol being self-enforcing and beneficial for the participants (a distributed reputation protocol must be co-utile for peers to rationally collaborate in the protocol). Section 3 states the requirements of a co-utile privacy-preserving reputation system. In Section 4 we recall the EigenTrust distributed reputation system, which meets many of our requirements. In Section 5, we describe a co-utile and weakly anonymous extension of EigenTrust. Then, in Section 6, we present a novel further extension, that also offers enhanced privacy to agents, both when their reputation is computed and when they help computing the reputations of other agents. Conclusions and future work directions are gathered in Section 7.

2 Co-utility

When setting forth for decentralized systems, peer-to-peer (P2P) architectures are the most appealing ones, because they empower individual agents. The challenge in P2P is how to ensure that peers will collaborate as expected. *Co-utility* [3, 4] is a specially attractive form of self-enforcing collaboration between agents. Co-utile protocols are those in which helping other (rational) agents increase their utilities is also the best way to increase one's own utility.

We can formalize the definition of co-utility in game-theoretic terms. A *game* is an abstraction of a scenario in which a set of rational agents can take decisions [7, 10]. We will focus on sequential perfect-information games, consisting of several rounds so that the agent choosing an action in a certain round knows the actions chosen in previous rounds (by others and herself). A game of this class can be represented in the so-called *extensive form*, that is a tree in which: (i) nodes are the points where decisions are made, (ii) each node is labeled with the name of the agent making the decision at that node, (iii) edges going out from a node represent the available actions that can be chosen at the node, and (iv) each of the terminal nodes (leaves) of the tree is labeled with the tuple of payoffs that agents obtain when the node is reached. Now we can give game-theoretic definitions of *protocol*, *self-enforcing protocol* and *co-utility.*

Definition 1 (Protocol). *Given a perfect-information game G in extensive form represented as a tree, a protocol is either a path from the root to a leaf or a subtree from the root to several leaves. In the latter case, alternative edges are labeled with probabilities of being chosen.*

Definition 2 (Self-enforcing protocol). *A protocol P on a game G is self-enforcing if no agent can increase her utility by deviating from P, provided that the other agents stick to P. Equivalently, at each successive node of the protocol path, sticking to the next action prescribed by the protocol (taking the next edge in the path) is an equilibrium of the remaining subgame of G (the subtree rooted at the current node). More technically, P on G is self-enforcing if and only if P is a subgame perfect equilibrium of G.*

Definition 3 (Co-utility). *A self-enforcing protocol P on a game G is co-utile if it is Pareto optimal and the utility derived by each participating agent is strictly greater than the utility the agent derives from not participating.*

A co-utile protocol P is Pareto-optimal in the sense that there is no alternative protocol P' giving greater utilities to all agents and strictly greater utility to at least one agent. See [3] for more details on co-utility.

3 Requirements of a co-utile and distributed reputation protocol

Many reputation mechanisms have been proposed in the literature for peer-to-peer communities, either centralized or distributed (see the surveys [5, 9]). Reputation is a very versatile artificial utility that can be used both as a reward and as a penalty. In fact, the reputation incentive can be added to a non-co-utile protocol to render it co-utile: it is a matter of rewarding with reputation increases those actions within the protocol and penalizing any deviation with reputation decreases. Yet, *the reputation calculation and management protocol* *itself should be naturally co-utile*, without requiring any additional incentives. Otherwise, computing reputations would not be rationally sustainable and would not serve its purpose of inducing co-utility or self-enforcement in other P2P protocols.

Specifically, we want a reputation protocol with the following features, which should make it amenable to co-utility, privacy-preserving and hence socially acceptable as a generalized reputation protocol:

- *Decentralization.* Reputations should be computed and enforced by the peers themselves rather than by a central authority. A central authority knows everything on everyone (no privacy) and is a single point of failure.
- *Privacy protection for those agents computing reputations and those whose reputation is computed.* See justification in Section 1 above. Surprisingly, most reputation schemes in the literature provide very little privacy or no privacy at all [11].
- *Low overhead.* The computational cost (bandwidth, storage, calculation) of computing reputations should be low (linear or quasi-linear). Otherwise, the negative utility of those costs might dominate the benefits brought by reputation.
- *Proper management of new agents.* Newcomers should not enjoy any reputation advantage. Otherwise, malicious peers may be motivated to create new anonymous identifiers after abusing the system in order to regain the advantages of a good reputation.
- *Attack tolerance.* A number of attacks may be orchestrated in order to subvert the reputation system. Since we assume rational players, we must make the cost of such attacks unattractively high. Following the classification in [5], we must avoid the following attacks: *self-promotion* (agents falsely increasing their reputations), *whitewashing* (creating new clean identities to get rid of bad reputations), *slandering* (falsely lowering other agents' reputations in order for one's own reputation to become comparatively higher) and *denial of service* (agents blocking the calculation and dissemination of reputation values).

After examining a number of decentralized reputation mechanisms [5], we selected EigenTrust [6] as a starting point to obtain a co-utile distributed reputation protocol. EigenTrust offers most of the desirable features identified above: distributed reputation calculation, low overhead and robustness to attacks. This reputation scheme is designed to filter out inauthentic content in peer-to-peer file sharing networks. Its basic idea is to calculate a global reputation for each agent based on aggregating the local opinions of the peers that have interacted with the agent. If we represent the local opinions by a matrix whose component (i, j) contains the opinion of agent i on agent j, the distributed calculation mechanism computes global reputation values that approximate the left principal eigenvector of this matrix. For unstructured networks, other solutions can be used, such as gossip-based reputation distribution protocols [13].

4 The EigenTrust distributed reputation system

In EigenTrust, a local reputation is assigned by an agent to another agent with whom the former has directly interacted, as a function of her opinion on the latter's behavior in the interaction. Then, the global reputation of an agent is computed in a distributed way, by means of a protocol whereby the agents share their local reputation values. The original EigenTrust system was designed for P2P file sharing, and the file receiver's opinion on a file download is just binary: satisfactory or not satisfactory. Since we want to be able to compute reputations based on opinions that have several categories or are even continuous, in [2] we extended EigenTrust to compute reputations based on non-binary opinions.

4.1 Computing local reputations

The opinion of an agent $\mathcal{P}_i$ on another agent $\mathcal{P}_j$ with whom $\mathcal{P}_i$ has directly interacted is the reputation s_{ij} of $\mathcal{P}_j$ local to $\mathcal{P}_i$. We define this value as the aggregation of payoffs (either positive or negative) that $\mathcal{P}_i$ has obtained from the set of transactions (Y_{ij}) performed with $\mathcal{P}_j$:

$$s_{ij} = \sum_{y_{ij} \in Y_{ij}} \mathit{payoff}_i(y_{ij}).$$

Payoffs may be binary (positive/negative opinion), discrete (an opinion from a discrete scale) or continuous (for example, the cost incurred in the transaction, in terms of bandwidth, time, etc.).

4.2 Computing global reputations

We review how EigenTrust computes global reputations from local reputations. First, in order to properly aggregate the local reputation values computed by peers, a normalized version c_{ij} of every local reputation s_{ij} assigned by peer $\mathcal{P}_i$ to any other peer $\mathcal{P}_j$ is computed as:

$$c_{ij} = \frac{\max(s_{ij}, 0)}{\sum_j \max(s_{ij}, 0)}.$$

The normalized local reputations lie between 0 and 1 and the sum of all normalized local reputations awarded by $\mathcal{P}_i$ to other peers is 1. In other words, each agent has a reputation budget of only 1 that she has to split among her peers proportionally to her positive experiences. In this way, all agents have the same influence on the global reputation. In particular, there is no dominance by the agents with more experiences and peers cannot collude by assigning arbitrarily high values to good peers. Regarding the truncation of negative values to 0, on the one hand it prevents $\mathcal{P}_i$ from assigning arbitrarily low values to other peers (thus neutralizing slandering), and on the other hand it sets newcomers on an equal footing with peers with whom $\mathcal{P}_i$ had a negative experience (thus neutralizing whitewashing).

The next step is to disseminate normalized local reputations and have them aggregated through the network peers by leveraging transitive reputation. Any agent $\mathcal{P}_i$ can compute $\hat{t}_{ik}^{(0)}$, an approximation of the reputation of a potentially unknown peer $\mathcal{P}_k$, by asking the peers with whom $\mathcal{P}_i$ has interacted ($\mathcal{P}_j$) for their local reputation w.r.t. $\mathcal{P}_k$, that is c_{jk}. Since $\mathcal{P}_i$ has already computed the local normalized reputation w.r.t. $\mathcal{P}_j$, that is c_{ij}, $\mathcal{P}_i$ can compute a local estimate of the reputation t_{ik} of $\mathcal{P}_k$ by using c_{ij} to weight $\mathcal{P}_j$'s local reputation; specifically, $\hat{t}_{ik}^{(0)} = \sum_j c_{ij} c_{jk}$. Thanks to the local normalization, $\hat{t}_{ik}^{(0)}$ takes values between 0 and 1. Observe that if we call $\mathbf{c}_i = (c_{i1}, \ldots, c_{in})^T$ and $\mathbf{C} = [c_{ij}]$, then $\hat{\mathbf{t}}_i^{(0)} = \mathbf{C}^T \mathbf{c}_i$, where $\hat{\mathbf{t}}_i^{(0)} = (\hat{t}_{i1}^{(0)}, \ldots, \hat{t}_{in}^{(0)})^T$. If every agent $\mathcal{P}_i$ computes $\hat{\mathbf{t}}_i^{(0)}$, in the next iteration $\mathcal{P}_i$ can compute $\hat{\mathbf{t}}_i^{(1)} = \mathbf{C}^T(\mathbf{C}^T \mathbf{c}_i)$. After m iterations, $\mathcal{P}_i$ will compute $\hat{\mathbf{t}}_i^{(m-1)} = (\mathbf{C}^T)^m \mathbf{c}_i$. Under the assumptions that $\mathbf{C}$ is irreducible and aperiodic [6], in an ideal and static setting the succession of reputation vectors computed by any peer will converge to the same vector for every peer, which we call $\mathbf{t} = (t_1, \ldots, t_n)^T$ and is the left principal eigenvector of $\mathbf{C}$. The j-th component of $\mathbf{t}$ represents the global reputation of the system on agent $\mathcal{P}_j$, for $j = 1, \ldots, n$.

Unfortunately, computing the global reputations by the above method is not efficient, because it takes too much communication. This and other issues are fixed in the proposals discussed in the next sections.

5 Co-utile and weakly anonymous computation of global reputations

In the same EigenTrust paper [6], the authors gave a secure version of the protocol to compute global reputations, in which not every peer contributed to computing the global reputation on every other peer. In [2], we extended that version to make it co-utile and ensure some level of anonymity, and we summarize this extension here.

In our extension, each agent $\mathcal{P}_i$ has an initial global reputation $t_i^{(0)}$ (based on previous experiences or assigned by default) and M score managers that will update her reputation value. Given a pseudonym ID_i of agent $\mathcal{P}_i$ (the pseudonym is $\mathcal{P}_i$'s identifier in the P2P network), her score managers are defined by a distributed hash table (DHT), which maps ID_i to M score managers whose pseudonyms are closest (according to an agreed distance), respectively, to values $h_0(ID_i)$, $\ldots$, $h_{M-1}(ID_i)$, where $h_0, h_1, \ldots, h_{M-1}$ are hash functions. The use of pseudonyms guarantees some level of anonymity (a weak level, as argued in the next section) and the use of hash functions prevents anyone from choosing a particular pseudonym as her score manager.

With the above arrangement, on average every agent is the score manager for M agents, so the work is balanced. Let D_i be the set of daughters of $\mathcal{P}_i$, that is, the set of agents for whom $\mathcal{P}_i$ is a score manager. During the computation of the global reputation, each $\mathcal{P}_i$ learns, for each $\mathcal{P}_d \in D_i$, the set A_d of agents that directly interacted with $\mathcal{P}_d$ (to provide or receive help) and receives the normalized local reputations c_{jd} on $\mathcal{P}_d$ from each $\mathcal{P}_j \in A_d$. The terms c_{jd} for $j \notin A_d$ are zero. Then $\mathcal{P}_i$ engages in an iterative refinement, for $k = 0, 1, 2, \ldots$:

$$t_d^{(k+1)} = c_{1d}t_1^{(k)} + c_{2d}t_2^{(k)} + \ldots + c_{nd}t_n^{(k)}. \tag{1}$$

Note that, in Expression (1), the weight attached to each normalized local reputation c_{jd} on $\mathcal{P}_d$ received from $\mathcal{P}_j$ is the current global reputation $t_j^{(k)}$ of $\mathcal{P}_j$. The refinement ends when the global reputation for $\mathcal{P}_d$ changes less than a small value $\epsilon > 0$ from k to $k + 1$. The resulting global reputations t_d are kept for a period until they are recomputed. The length of the reputation update period is a parameter of the system.

With this arrangement, not only the entire computation is mediated by the score managers, but also the dissemination of global reputations. For any $\mathcal{P}_i$, her global reputation t_i can be obtained from her score managers.

Briefly speaking, *this protocol to compute global reputations is co-utile because it encourages the agents to collaborate.* The reason is that the impact of their opinions on the computation of the global reputations increases when they are active (that is, when they are members of as many sets A_* as possible). See [2] for a detailed co-utility analysis.

6 Co-utile and privacy-preserving computation of global reputations

The protocol in Section 5 provides only weak anonymity, because it uses a single pseudonym for each agent. Indeed, if all the interactions of an agent $\mathcal{P}_i$ are carried out under the same pseudonym, in the end the agent can be identified from its graph of interactions alone. This is well known in graph anonymization, where it is clear that just replacing the node labels with pseudonyms is not a sufficient anonymization (see, *e.g.* [8]).

A possible way to improve the privacy of agents is to split the identity of each agent into as many pseudonyms as the agent wishes. We consider two cases: multiple pseudonyms with independent reputations and multiple linkable pseudonyms.

6.1 Multiple pseudonyms with independent reputations

This case makes sense if it is socially tolerable that an agent leads several parallel lives each with its own independent reputation. Splitting into parallel lives is not necessarily unfair because:

- If an agent behaves very well under a pseudonym and earns a high reputation with that pseudonym, she will only enjoy the benefits of high reputation under that pseudonym.
- On the other hand, if the agent behaves very poorly under another pseudonym, she will suffer the effects of a bad reputation only under that pseudonym.

In practice, this scenario amounts to splitting an agent into several surrogate agents. So instead of agent $\mathcal{P}_i$ having a single pseudonym ID_i, she will be allowed to have several pseudonyms $ID_i^1, ID_i^2, \ldots$, etc. Otherwise, the scheme will work exactly as described in Section 5.

6.2 Multiple linkable pseudonyms

In this case, the agent can lead several independent lives under different pseudonyms. However, at any point she can choose to link some of her pseudonyms and thereby merge their corresponding reputations. This makes sense *if the motivation of the agent when using several pseudonyms is to improve her privacy, but she would like to enjoy the same (or a similar) reputation no matter which among her pseudonyms she is using.* To allow this, the following protocols can be followed.

We assume $\mathcal{P}_i$ wants to create k pseudonyms that are linkable to each other at a later time. Protocol 1 creates multiple linkable pseudonyms.

Protocol 1 (Linkable pseudonym creation)

1. *$\mathcal{P}_i$ chooses strings ID_i^1, ID_i^2, …, ID_i^k.*
2. *$\mathcal{P}_i$ computes*

$$\begin{aligned} H(ID_i^1 || H_i^1, K_i^1) &= R_i^1 \\ H(ID_i^2 || H_i^2, K_i^2) &= R_i^2 \\ \vdots \; &\vdots \; \vdots \\ H(ID_i^k || H_i^k, K_i^k) &= R_i^k \end{aligned} \tag{2}$$

   where $||$ is the concatenation operator, $H(\cdot, \cdot)$ is a secure keyed hash function,

$$H_i^j = H'(ID_i^1 || \ldots || ID_i^{j-1} || ID_i^{j+1} || \ldots || ID_i^k), \quad j = 1, \ldots, k$$

   with $H'(\cdot)$ being a secure hash function, and K_i^1, …, K_i^k are random keys. The values H_i^j and the keys K_i^j are known only to $\mathcal{P}_i$ for $j = 1, \ldots, k$.
3. *$\mathcal{P}_i$ can now operate under $ID_i^1 || R_i^1$, $ID_i^2 || R_i^2$, …, $ID_i^k || R_i^k$ as her pseudonyms.*

Linkable pseudonyms are longer than normal pseudonyms, because they are appended the hash image as a suffix. Although their random-looking suffix makes them distinguishable from normal pseudonyms, this should not be a problem as long as a sufficient number of agents is using linkable pseudonyms. However, if a single agent (or very few agents) used them, the linkable pseudonyms could be easily linked by anyone, and the anonymity gain of the agent's having several pseudonyms would be canceled.

If, at a certain point of time, $\mathcal{P}_i$ wants to link some of her linkable pseudonyms, say without loss of generality, $ID_i^1 || R_i^1$, $ID_i^2 || R_i^2$, …, $ID_i^{k'} || R_i^{k'}$, for $k' \leq k$, she can do so using the following protocol.

Protocol 2 (Pseudonym linkage)

1. *$\mathcal{P}_i$ sends $ID_i^1||R_i^1$, …, $ID_i^{k'}||R_i^{k'}$, K_i^1, …, $K_i^{k'}$, H_i^1, …, $H_i^{k'}$ to the score managers corresponding to the k' pseudonyms to be linked, that is, to the M score managers whose pseudonyms are closest to $h_0(ID_i^1||R_i^1), \ldots, h_{M-1}(ID_i^1||R_i^1)$, and to the M score managers whose pseudonyms are closest to $h_0(ID_i^2||R_i^2)$, $\ldots, h_{M-1}(ID_i^2||R_i^2)$, and so on, up to the M score managers whose pseudonyms are closest to $h_0(ID_i^{k'}||R_i^{k'})$, $\ldots$, $h_{M-1}(ID_i^{k'}||R_i^{k'})$.*
2. *The score managers check that the k' pseudonyms are currently marked as unlinked. Any pseudonyms that have been linked in previous instances of the protocol are discarded. Without loss of generality, assume that the first k'' pseudonyms are unlinked, with $k'' \leq k'$.*
3. *The score managers verify whether the following equations hold:*

$$\begin{aligned} H(ID_i^1||H_i^1, K_i^1) &\stackrel{?}{=} R_i^1 \\ H(ID_i^2||H_i^2, K_i^2) &\stackrel{?}{=} R_i^2 \\ \vdots \;\vdots &\;\vdots \\ H(ID_i^{k''}||H_i^{k''}, K_i^{k''}) &\stackrel{?}{=} R_i^{k''} \end{aligned} \tag{3}$$

   where the j-the equation is verified by the M score managers managing $ID_i^j||R_i^j$, for $j = 1, \ldots, k''$.
4. *If all checks in Expression (3) hold, the score managers of $ID_i^j||R_i^j$, for $j = 1, \ldots, k''$, will consider the k' pseudonyms as linked from now on. If only a subset of at least two equations holds, then the score managers will consider as linked only the subset of pseudonyms corresponding to that subset. If only one equation holds, no pseudonyms will be linked. The implications of having a set of linked pseudonyms are:*
   (a) *The newly linked pseudonyms are marked by the corresponding score managers as "linked".*
   (b) *Each time the global reputation of one of the linked pseudonyms is updated:*
      i. *The score managers of that pseudonym notify the pseudonym's updated reputation to the score managers of the other linked pseudonyms.*
      ii. *The score managers of all linked pseudonyms recompute the reputation of the linked pseudonyms as the aggregation of the reputations individually earned by those pseudonyms (using any agreed aggregation operator, like for example the mean).*
      iii. *The score managers replace the reputations of the linked pseudonyms with the newly computed aggregated reputation. A small random perturbation can be added to the aggregated reputation of each pseudonym, to avoid that exactly equal reputations leak to everyone that pseudonyms are linked.*

An interesting point of Protocol 2 is that, when $\mathcal{P}_i$ wants to link only a strict subset of her pseudonyms ($k' < k$), the rest of her pseudonyms are not disclosed by the protocol as belonging to $\mathcal{P}_i$.

Note 4 (On the co-utility of multiple linkable pseudonyms). The extension to multiple linkable pseudonyms presented in this section is in the interest of agent $\mathcal{P}_i$. Since Protocol 1 involves only $\mathcal{P}_i$, it does not endanger the co-utility of the protocol previous to the extension (justified in [2]). On the other hand, Protocol 2 involves $\mathcal{P}_i$ and the score managers. The motivation of $\mathcal{P}_i$ to adhere to the protocol is clear. Regarding the score managers, they also can be assumed to follow the protocol in their own interest, because the availability of accurate reputations is good for all peers (the score managers do not deviate for the same reasons they did not deviate from the protocol previous to the extension, see the co-utility analysis in [2]).

Note 5 (On disagreements between score managers). We have argued in the previous note that rationally selfish score managers will adhere to Protocol 2. However, some score managers might be not only selfish, but interested in malicious deviation (*e.g.* they may be offered money to alter a certain agent's reputation). Yet, consistently with the EigenTrust security model, we assume that a majority among the M score managers assigned to each agent are honest (even if rationally selfish), so that the reputation value reported by the majority can be assumed to be correct. If small random perturbations are added to differentiate the aggregate reputation reported for each linked pseudonym (as suggested at the end of Protocol 2), only those reputation differences beyond the perturbation range will be regarded as disagreements.

Note 6 (Confidentiality of pseudonym linkage). When several pseudonyms are linked, the linkage becomes known to the score managers of the linked pseudonyms, but, unless those managers tell other peers, no one else needs to know about the linkage. Hence, linked pseudonyms are better in terms of privacy than a single pseudonym replacing all of them.

Proposition 7 (Security of pseudonym linkage). *Only the agent who created a pseudonym can link it to other pseudonyms. Given a pseudonym created by an agent $\mathcal{P}_i$, no other agent $\mathcal{P}_j$ can create and link a pseudonym to a pseudonym of $\mathcal{P}_i$ without the latter's consent.*

Proof. Non-linkable pseudonyms do not need to be considered, because they lack the hash images R needed for the score managers to verify the linkage.

For a linkable pseudonym $ID_i^j||R_i^j$ to be linked to other pseudonyms, the score managers of the pseudonym need to be provided with K_i^j and H_i^j, so that they can verify the j-th equation in Expressions (3). But, if $ID_i^j||R_i^j$ is still unlinked, K_i^j is only known to the agent $\mathcal{P}_i$ who created the pseudonym; this is ensured by the security of the keyed hash function employed in Expressions (2). On the other hand, if $ID_i^j||R_i^j$ is already linked, by design of Protocol 2 no one can link it again.

Regarding the second statement, agent $\mathcal{P}_l$ can fabricate a pseudonym $ID_l^{j'}||R_l^{j'}$ with the aim of linking it to a pseudonym $ID_i^j||R_i^j$ created by agent $\mathcal{P}_i$. In order for the two pseudonyms to be considered as linked, the score managers for

$ID_l^{j'}||R_l^{j'}$ should receive $K_l^{j'}$ and $H_l^{j'}$ such that $H(ID_l^{j'}||H_l^{j'}, K_l^{j'}) = R_l^{j'}$, and the score managers for $ID_i^j||R_i^j$ should receive K_i^j and H_i^j such that $H(ID_i^j||H_i^j, K_i^j) = R_i^j$. Now, $\mathcal{P}_l$ can choose any $H_l^{j'}$ and $K_l^{j'}$ and take $R_l^{j'} := H(ID_l^{j'}||H_l^{j'}, K_l^{j'})$. However, regarding $ID_i^j||R_i^j$, only $\mathcal{P}_i$ knows K_i^j if the pseudonym is unlinked; if it was already linked, it cannot be linked again. □

6.3 Generalization: pseudonyms allowing multiple linkages

In the protocols described in Section 6.2, a linkable pseudonym can only be linked once. This is because there is a single secret key K_i^j for a pseudonym ID_i^j: after the agent $\mathcal{P}_i$ owning the pseudonym discloses K_i^j to perform the linkage in Protocol 2, the owner would no longer be the only one able to link the pseudonym if further linkages were allowed.

A way to overcome this problem is to create linkable pseudonyms with several secret keys. For example, to allow up to ℓ linkages, $\mathcal{P}_i$ could create and disseminate a pseudonym as: $ID_i^j||R_i^{j,1}, \ldots, R_i^{j,\ell}$ where

$$H(ID_i^j||H_i^j, K_i^{j,1}) = R_i^{j,1} \tag{4}$$

$$H(ID_i^j||H_i^j, K_i^{j,2}) = R_i^{j,2} \tag{5}$$

$$\vdots \quad \vdots \quad \vdots$$

$$H(ID_i^j||H_i^j, K_i^{j,\ell}) = R_i^{j,\ell}.$$

In this way, there are ℓ secret keys. Up to ℓ linkages of the pseudonym can be performed by using a slightly generalized version of Protocol 2, where in the first linkage $\mathcal{P}_i$ would reveal $K_i^{j,1}$ that satifies Equation (4), in the second linkage $\mathcal{P}_i$ would reveal $K^{j,2}$ that satisfies Equation (5), and so on. The score managers for the pseudonym would maintain a counter with the number of times the pseudonym has been linked, and they would not accept the same key more than once.

7 Conclusions and future work

We have presented a peer-to-peer fully distributed reputation system that is privacy-preserving, in that it allows peers to use any number of pseudonyms and link some of them if they want to enjoy the same (or a similar) reputation under several of their pseudonyms. A reputation system of this kind is a better candidate than a centralized reputation system for generalized use in a social credit system.

Future research will be devoted to improving the management of pseudonyms. Specifically, we will investigate solutions that improve the confidentiality of pseudonym linkage. Also, allowing multiple linkages of a pseudonym without expanding its length deserves further work. Another interesting direction is to create revocable pseudonyms that can be attributed to a certain agent with the help of a trusted third party.

Acknowledgments and disclaimer

Partial support to this work has been received from the Templeton World Charity Foundation (grant TWCF0095/AB60 "CO-UTILITY"), ARC (grant DP160100913), the European Commission (projects H2020-644024 "CLARUS" and H2020-700540 "CANVAS"), the Government of Catalonia (ICREA Acadèmia Prize) and the Spanish Government (projects TIN2014-57364-C2-1-R "SmartGlacis" and TIN 2015-70054-REDC). The author holds the UNESCO Chair in Data Privacy, but the views in this paper are the author's own and are not necessarily shared by UNESCO.

References

1. R. Creemers. *China Copyright and Media*. https://chinacopyrightandmedia.wordpress.com/about/. Checked December 31, 2017.
2. J. Domingo-Ferrer, O. Farràs, S. Martínez, D. Sánchez, J. Soria-Comas. Self-enforcing protocols via co-utile reputation management. Information Sciences 367-368 (2016) 159-175.
3. J. Domingo-Ferrer, S. Martínez, D. Sánchez, J. Soria-Comas. Co-utility: self-enforcing protocols for the mutual benefit of participants. Engineering Applications of Artificial Intelligence 59 (2017) 148-158.
4. J. Domingo-Ferrer, D. Sánchez, J. Soria-Comas. Co-utility - self-enforcing collaborative protocols with mutual help. Progress in Artificial Intelligence 5(2) (2016) 105-110.
5. K. Hoffman, D. Zage, C. Nita-Rotaru. A survey of attack and defense techniques for reputation systems. ACM Computing Surveys 42(1) (2009) art. no. 1.
6. S. D. Kamvar, M. T. Schlosser, H. Garcia-Molina. The EigenTrust algorithm for reputation management in P2P networks. In: Proceedings of the 12th International Conference on World Wide Web, ACM, 2003, pp. 640-651.
7. K. Leyton-Brown, Y. Shoham. Essentials of Game Theory: A Concise, Multidisciplinary Introduction, Morgan & Claypool, 2008.
8. K. Liu, E. Terzi. Towards identity anonymization on graphs. In: Proceedings of the 2008 ACM SIGMOD Intl. Conf. on Management of Data - SIGMOD'08, ACM, 2008, pp. 93-106.
9. S. Marti, H.Garcia-Molina. Taxonomy of trust: categorizing P2P reputation systems. Computer Networks 50(4) (2006) 472-484.
10. M. Osborne, A. Rubinstein. A Course in Game Theory, MIT Press, 1994.
11. A. Singh, L. Liu. TrustMe: anonymous management of trust relationships in decentralized P2P systems. In: Proceedings of the Third International Conference on Peer-to-Peer Computing (P2P 2003), 2003, 142-149.
12. S. Hsu. China's new social credit system. The Diplomat, May 10, 2015. http://thediplomat.com/2015/05/chinas-new-social-credit-system/
13. R. Zhou, K. Hwang, M. Cai. GossipTrust for fast reputation aggregation in peer-to-peer networks. IEEE Transactions on Knowledge and Data Engineering 20(9) (2008) 1282-1295.