Anonymity and Rewards in Peer Rating Systems

Lydia Garms✉[1]*, Siaw–Lynn Ng[1], Elizabeth A. Quaglia[1], and Giulia Traverso[2]

[1] Royal Holloway, University of London, UK
{Lydia.Garms,S.Ng,Elizabeth.Quaglia}@rhul.ac.uk
[2] Cysec, Lausanne, Switzerland
giulia.traverso@cysec.systems

Abstract. When peers rate each other, they may rate inaccurately to boost their own reputation or unfairly lower another's. This could be mitigated by having a reputation server incentivise accurate ratings with a reward. However, assigning rewards becomes challenging when ratings are anonymous, since the reputation server cannot tell which peers to reward for rating accurately. To address this, we propose an anonymous peer rating system in which users can be rewarded for accurate ratings, and we formally define its model and security requirements. In our system ratings are rewarded in batches, so that users claiming their rewards only reveal they authored one in this batch of ratings. To ensure the anonymity set of rewarded users is not reduced, we also split the reputation server into two entities, the Rewarder, who knows which ratings are rewarded, and the Reputation Holder, who knows which users were rewarded. We give a provably secure construction satisfying all the security properties required. For our construction we use a modification of a Direct Anonymous Attestation scheme to ensure that peers can prove their own reputation when rating others, and that multiple feedback on the same subject can be detected. We then use Linkable Ring Signatures to enable peers to be rewarded for their accurate ratings, while still ensuring that ratings are anonymous. Our work results in a system which allows accurate ratings to be rewarded, whilst still providing anonymity of ratings with respect to the central entities managing the system.

1 Introduction

Anonymity has long been a sought-after property in many cryptographic primitives, such as public-key encryption [5], identity-based encryption [2, 15], and a defining one in others, such as group signatures [16] and ring signatures [46]. A plethora of more complex protocols, from broadcast encryption [35] to cryptocurrencies [32], have been enhanced by user anonymity.

An example of such protocols are *rating systems*, also referred to as reputation systems, in which users can be rated by providing feedback on goods or services, with the support of a reputation server. Each user has a reputation value based

* The author was supported by the EPSRC and the UK government as part of the Centre for Doctoral Training in Cyber Security at Royal Holloway, University of London (EP/K035584/1) and by the InnovateUK funded project AQuaSec.

on these ratings, which can be used to evaluate their trustworthiness. In this context, the value of anonymity lies in the fact that users are able to give honest feedback without fear of repercussions. This may occur when there is a lack of trust for the reputation server, or when users are concerned about retaliation.

Anonymity has received a great amount of attention in this area and abundant existing literature covers a range of anonymous rating systems in both the centralised and distributed settings. Distributed systems, e.g., [38], have no reputation server and use local reputation values, i.e., reputation values created by users on other users. For example, a user may generate a reputation value based on feedback from querying other users. This means a user does not have a unique reputation value, but many other users hold their own reputation value for them. In this setting, privacy preserving decentralised reputation systems [43] are designed to maintain anonymity when answering queries from other users.

We focus on centralised systems, since the reputation systems used by most service providers such as Airbnb, Uber and Amazon are of this type. In the centralised setting, a central reputation server enrols users and forms reputation values on these users. In [11, 23, 10, 18, 17] anonymity of a rating is provided to all except the reputation server, and multiple ratings cannot be given on the same subject. In [51], multiple reputation servers are used so that anonymity of ratings holds, unless all reputation servers collude. Other works provide anonymity of ratings in the presence of a corrupted reputation server [47, 44, 25, 27]. In [3, 8] anonymity is achieved with a different approach. The reputation server still enrols users, but no longer forms reputations. Instead users collect tokens based on anonymous ratings from other users and prove their own reputation.

Whilst the benefits of anonymity are clear, it is also understood that this same property can provide an opportunity for malicious users to misbehave. They may "bad mouth" other users, for instance competitors, giving dishonest negative feedback to these users to decrease their reputation. Or they may collude and give each other positive feedback in order to inflate their own reputation. To avoid this, the system can provide either a mechanism to revoke the malicious user's anonymity (typically achieved through a traceability property), or incentivize good behaviour by rewarding users. The rating systems proposed so far approach this issue via user tracing. Indeed, in schemes where the reputation server can de–anonymise ratings [11, 23, 18, 17], inaccurate ratings can be punished.

We take a different approach by rewarding honest ratings in *anonymous peer rating systems*, where users are peers and anonymously rate each other. Examples include peer-to-peer file sharing [1], collaborative knowledge production [40, 14, 28], and shared knowledge of internet and software vulnerabilities [30, 48]. In such systems the rewarding approach works well since raters are also participating within the system and so have an interest in rating accurately to increase their reputation through rewards. The use of incentives to encourage accurate feedback has already been discussed in [42, 50], but ratings are not anonymous.

Privacy-preserving incentive schemes [39, 33, 29, 9, 12], where users can be incentivised anonymously without their transactions being linked, have also been proposed. In [39] it is described how such incentives could contribute towards a

reputation value. However, these schemes do not capture the ability to reward accurate ratings. Firstly, ratings must be incentivised as they are submitted, at which point it is not known whether a rating is accurate. When accurate ratings are determined it is then difficult to return the incentive to the relevant user. Secondly, in [33, 29, 9, 12], a user's balance is updated each time a user receives an incentive. However, a user may have submitted k accurate rating on other users, which are unlinkable. Then their balance of n should increase by k, but instead they receive k updated tokens for a balance of $n + 1$. Finally, in [39, 29, 9, 12] a user would have to participate in an interactive protocol to rate others.

Therefore the challenge remains to rewards users that rate accurately, whilst preserving the anonymity of their ratings even with respect to the reputation server. This is what we address in this paper.

1.1 Our work

We consider an anonymous peer rating system in which, at each round of interaction, users rate each other by providing feedback to the reputation server.

Our contribution is to allow accurate ratings to be incentivised and weighted by reputation, whilst still ensuring anonymity of ratings. Achieving this is challenging for two reasons. First, the reputation used to weight feedback could be used to de-anonymise a user. We can partially mitigate this by ensuring reputation is coarse-grained as in [8] (by rounding the reputation value, for instance), which ensures that a user who has a unique reputation score does not reveal their identity. The trade off between precision of reputation and size of anonymity sets is further discussed in [45]. Second, and crucially, accurate ratings must be incentivised without being de-anonymised. We achieve this by incentivising a large set of ratings simultaneously, and rewarding the users responsible for such ratings. With this approach, however, the anonymity set can be reduced substantially. Indeed, a malicious reputation server could decide to only reward a small number of ratings it seeks to de-anonymise, and then check which users are rewarded with an increase in reputation. These users then must have authored these ratings.

A way to lessen the impact in both cases is to restrict access to reputation. A specific trusted entity, the Reputation Holder, holds the reputations of users, and the latter should only be revealed sparingly. We do not specify exactly when and how reputations should be revealed in order to allow for a flexible scheme, and because this has been discussed in the existing literature. For example, in [27, 47], users can prove their reputation and so can decide which users to reveal it to. A simpler example is that a user would have to demonstrate a good reason to learn another's reputation from the Reputation Holder.

We go further and introduce a new entity, the Rewarder, who chooses which ratings to reward, and who cannot see which users have their reputation increase. As the Reputation Holder no longer knows which ratings were rewarded, they cannot compare these ratings with the users that claim rewards and so reduce the anonymity set. We formalise this in the Anonymity of Ratings under a Corrupt Reputation Holder requirement. For completeness, we also consider the case that the Reputation Holder and the Rewarder collude or are the same entity. Clearly

they learn that each user that was rewarded n times, authored n of the ratings rewarded, however they should learn no more than this. We formalise this in our Anonymity of Ratings under Full Corruption requirement.

Although we are aware that using reputation values and incentivising accurate ratings both inescapably reduce the anonymity sets of ratings, in this work we aim to provide the best anonymity achievable given the functionality. Furthermore, we also must ensure that users do not attempt to subvert the system by claiming rewards that they are not entitled to, by providing multiple ratings on the same user per round, by lying about their reputation, or by framing other users so that they seem to be cheating. We formalise this in our Fair Rewards, Traceability[3], Unforgeability of Reputation and Non–Frameability requirements.

In this work we first provide a model and security requirements for an anonymous peer rating system APR, which formalises the necessary privacy and security properties discussed above. We use property-based definitions, which are intuitive and useful when proving security. We then give a construction that is provably secure given these security requirements. Our construction makes use of Direct Anonymous Attestation (DAA) [13], which we use to sign feedback. This ensures that, whilst signed feedback are unlinkable, multiple feedback on the same user can be detected, due to the user controlled linkability feature of DAA. We modify the DAA scheme so that when giving feedback a user can prove they have a particular reputation for that round, so that feedback can be weighted. We then make use of Linkable Ring Signatures [36] to allow to incentivise users who rate accurately. For every rating a freshly generated verification key is attached, encrypted under the Rewarder's public key. When the Rewarder rewards a rating, they publish the corresponding decrypted verification keys. The user can then sign a linkable ring signature with the corresponding secret key and claim their incentive from the Reputation Holder. The linkability of the signature scheme can be used to detect if a user tries to claim for the same incentive twice, whilst its anonymity ensures that ratings remain anonymous.

Although DAA and Linkable Ring Signature schemes are similar primitives, we note that they have subtly different properties that make them exactly suited to their particular role in building an APR scheme. As ring signature key pairs can be generated without involving any central entity, this allows a new verification key to be generated for every rating. The fact that a central entity, in our case the Reputation Holder, must authorise the creation of a new DAA key pair, prevents sybil attacks. Otherwise, users could easily create multiple identities and rate other users as many times as they wish per round. Unlike group signatures [16], DAA schemes do not allow a trusted opener to de–anonymise signatures, ensuring that anonymity of ratings holds with respect to the Rewarder.

While the main aim of our anonymous peer rating system is to ensure anonymous and honest feedback, it is also important to consider how it is affected by many other conventional attacks on rating systems. The unfair ratings attack [21] is mitigated by the detection of multiple ratings per subject per round. The in-

[3] Traceability here refers to the requirement that multiple ratings cannot be given on the same subject per round.

centives also encourage users to give more accurate feedback. The self–rating or self–promoting attack [31] is mitigated by encouraging all users to give feedback on their own performance. Sybil attacks [22], where a user creates multiple identities to join the system to give unfair feedback, can be mitigated by making joining the system expensive, and by a robust registration process. This also mitigates against whitewashing attacks [19], where a user leaves and rejoins to shed a bad reputation. The on-off attack [49], where a user behaves honestly to increase their reputation before behaving dishonestly, can be somewhat mitigated by adjusting the weighting of the final reputation formation in our system, so that bad behaviour will cause the reputation to deteriorate quickly. Reputation lag exploitation [34], where a user exploits the interval before the latest round of ratings takes effect, cannot be prevented but, as before, we can mitigate it by making the reputation deteriorate faster on bad behaviour.

2 Anonymous Peer Rating Systems: Definitions and Security Models

In this section, we introduce and formally define an anonymous peer rating (APR) system, and the security and privacy properties it should satisfy. We consider a set of users $\mathcal{U} = \{uid_i\}$ interacting with each other in rounds. At the end of each round they rate each other's performance, by anonymously sending a numerical feedback alongside their reputation to the Rewarder. The Rewarder collects ratings, discards multiple ratings on the same subject, and rewards accurate feedback by outputting a set of incentives. A user claims to the Reputation Holder that they were responsible for a number of these incentives. The final reputation held by the Reputation Holder on a user is based on three components: weighted feedback from other users, the number of incentives they have successfully claimed, and their previous reputation. We present an illustration of our model in Figure 1 and formally capture this as follows.

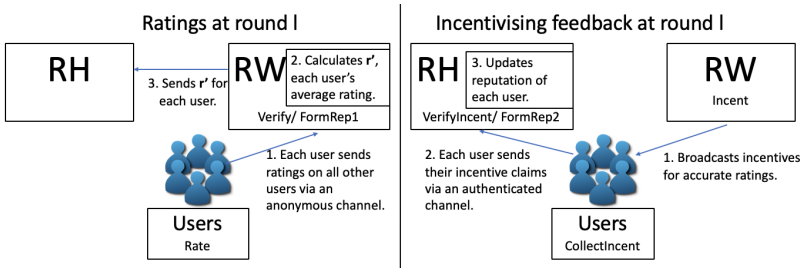

Fig. 1. Diagram illustrating our model.

Setup & Key Generation The Reputation Holder and Rewarder generate their own key pairs. The group public key $gpk = (\mathsf{param}, rwpk, rhpk)$ consists of the public keys of both entities.

Setup$(1^\tau, f_1, f_2) \rightarrow$ **param:** input a security parameter 1^τ, and two generic functions f_1 and f_2 which calculate the reputations of users. The function f_1 is input the number of ratings a user is being rewarded for, and outputs the second component, $\mathbf{r}''$, of their reputation for this round. The function f_2 is input the two components of a user's reputation for this round, and their reputation from the previous round, and outputs their final reputation for this round[4]. Setup outputs the public parameters param which include f_1, f_2.

RHKeyGen(**param**) $\rightarrow (rhsk, rhpk)$**:** performed by the Reputation Holder, outputs the Reputation Holder's secret key $rhsk$ and public key $rhpk$.

RWKeyGen(**param**) $\rightarrow (rwsk, rwpk)$**:** performed by the Rewarder, outputs the Rewarder's secret key $rwsk$ and public key $rwpk$.

Join When a user joins the system they engage in an interactive protocol with the Reputation Holder after which they are issued with secret keys used to provide anonymous ratings and to collect rewards for giving honest feedback. We assume users must join the system before a round of ratings begins.

$\langle$**Join**(gpk), **Issue**$(rhsk, gpk)\rangle$**:** a user uid joins the system by engaging in an interactive protocol with the Reputation Holder. The user uid and Reputation Holder perform algorithms Join and Issue respectively. These are input a state and an incoming message M_{in}, and output an updated state, an outgoing message M_{out}, and a decision, either cont, accept, or reject, which denote whether the protocol is still ongoing, has ended in acceptance or has ended in rejection respectively. (States are values necessary for the next stage of the protocol.) The initial input to Join is the group public key, gpk, whereas the initial input to Issue is the Reputation Holder's secret key, $rhsk$, and the group public key gpk. If the user uid accepts, Join privately outputs the user's secret key $\mathbf{gsk}[uid]$, and Issue outputs $\mathbf{reg}[uid]$, which stores the user's registration and will be used to later allocate that user a reputation.

Ratings at Round l Each user uid has a reputation $\mathbf{r}[uid, l]$ at round l, also held by the Reputation Holder. We assume that reputation is coarse-grained, which lessens the impact on anonymity with respect to the Reputation Holder. At Round l, a user uid with reputation r forms a rating ρ with Rate on user uid' based on a numerical feedback fb, which is sent to the Rewarder via a secure anonymous channel[5]. For flexibility we do not specify the form of fb, in [50] this a real number between 0 and 1. The user stores a trapdoor td for each rating for later use when claiming incentives. The Rewarder can verify ratings with Verify.

After collecting the valid ratings weighted by reputation, the Rewarder calculates an intermediate value $\mathbf{r}'[uid, l]$ for each uid with FormRep1, through which

[4] For example, in [50], f_1 is simply the number of incentives received multiplied by some weight, and f_2 is the weighted sum of these components.

[5] We require a secure channel to prevent the Reputation Holder from accessing the ratings, and determining which ratings will be rewarded by following the strategy of the Rewarder. This knowledge would allow the Reputation Holder to decrease the anonymity set of the users claiming incentives, as in the case when both the Rewarder and Reputation Holder are corrupted.

it also detect multiple ratings on the same subject. This value captures the average feedback given on uid weighted by the reputation of the rater, and is sent to the Reputation Holder via a secure authenticated channel.

Rate($\mathbf{gsk}[uid], gpk, fb, uid', l, r, \omega) \rightarrow (\rho, td)$: performed by the user with identifier uid, with input the user's secret key $\mathbf{gsk}[uid]$, the group public key gpk, a feedback fb, the user who they are rating uid', the current round l, their reputation r, and a reputation token ω output in the previous round by AllocateRep. Outputs a rating ρ and a trapdoor td.

Verify$(fb, uid', l, r, \rho, gpk) \rightarrow \{0, 1\}$: a public function that is performed by the Rewarder when receiving a rating tuple (fb, uid', r, ρ). Outputs 1 if ρ is valid on the feedback fb for user uid' at round l for reputation r under the group public key gpk, and 0 otherwise.

FormRep1$(uid, l, (fb_1, r_1, \rho_1), \cdots (fb_k, r_k, \rho_k), gpk) \rightarrow \mathbf{r}'[uid, l]$: performed by the Rewarder with input k valid rating tuples $\{(fb_i, uid, r_i, \rho_i) : i \in [1, k]\}$ on user uid at round l, and the group public key gpk. Outputs $\mathbf{r}'[uid, l] = \frac{\sum_{i=1}^{k} r_i fb_i}{\sum_{i=1}^{k} r_i}$ if all ratings originate from different users' secret keys. Otherwise outputs $\perp$ (in practice also outputs ratings that should be discarded).

Incentivising accurate feedback The Rewarder compares each feedback on uid'. If this is close to $\mathbf{r}'[uid', l]$ then this rating will be considered to be accurate and will be given an incentive. We define accurate as close to $\mathbf{r}'$. However, our model could simply be adapted to incorporate different metrics of accuracy.

The Rewarder inputs the k accurate ratings in this round to Incent, which outputs k incentives which are broadcast publicly to all users. Incent must be deterministic, to allow users to identify which incentives match their ratings.

A user collects all its incentives and can then use CollectIncent, along with the trapdoors stored earlier, to output an incentive claim σ for each of their incentives. They send these incentive claims to the Reputation Holder over a secure authenticated channel. Incentive claims are verified by the Reputation Holder with VerifyIncent. After gathering all the valid incentive claims, the Reputation Holder calculates the second component $\mathbf{r}''[uid, l+1]$ of a user's reputation at round l with FormRep2, which also checks that no user has claimed the same incentive twice. This value reflects how in line the feedback of uid is with respect to other users' feedback, incentivising users to give honest feedback.

Incent$((fb_1, uid_1, r_1, \rho_1), \cdots, (fb_k, uid_k, r_k, \rho_k), l, rwsk, gpk) \rightarrow t_1, \cdots, t_k$: a deterministic function performed by the Rewarder on input k rating tuples $\{(fb_i, uid_i, r_i, \rho_i) : i \in [1, k]\}$ from round l and its secret key $rwsk$. Outputs k incentives $t_1, \cdots, t_k$.

CollectIncent$(uid, (fb, uid', l, r, \rho, td), t_1, \cdots, t_k, gpk) \rightarrow \sigma$: performed by the user uid who gave the rating tuple (fb, uid', r, ρ) for round l corresponding to trapdoor td, with input the incentives output by the Rewarder $t_1, \cdots, t_k$. Outputs an incentive claim σ if the rating tuple (fb, uid', r, ρ) corresponds to an incentive in list $t_1, \cdots, t_k$ and $\perp$ otherwise.

VerifyIncent$(uid, \sigma, t_1, \cdots, t_k, gpk) \rightarrow \{0,1\}$**:** performed by the Reputation Holder when receiving an incentive claim σ from user uid on incentives $t_1, \cdots, t_k$. Outputs 1 if the incentive claim is valid on $uid, t_1, \cdots t_k$ and 0 otherwise.

FormRep2$(uid, \sigma_1, \cdots \sigma_{k_1}, t_1, \cdots, t_{k_2}, gpk) \rightarrow \mathbf{r}''[uid, l]$**:** performed by the Reputation Holder with input a user uid, k_1 valid incentive claims $\sigma_1, \cdots \sigma_{k_1}$ and k_2 incentives $t_1, \cdots, t_{k_2}$. Outputs $\mathbf{r}''[uid, l] = f_1(k_1)$ if no incentive has been claimed twice, and otherwise $\perp$.

Allocate reputation for next round For the first round, all users' reputations are set to an initial value. The reputation of user uid for round $l+1$, $\mathbf{r}[uid, l+1]$, is set by the Reputation Holder as $f_2(\mathbf{r}'[uid, l], \mathbf{r}''[uid, l], \mathbf{r}[uid, l])$ combining the user's previous reputation and the two intermediate values $\mathbf{r}'[uid, l], \mathbf{r}''[uid, l]$. This reputation value $\mathbf{r}[uid, l+1]$, which we refer to as r, and a reputation token ω obtained from AllocateRep are given to the user via a secure authenticated channel to allow them to prove they have this reputation in the next round.

AllocateRep$(uid, r, l, rhsk, \mathbf{reg}) \rightarrow \omega$**:** performed by the Reputation Holder with input a user uid with reputation r during round l, the Reputation Holder's secret key $rhsk$ and the registration table $\mathbf{reg}$. Outputs reputation token ω.

2.1 Security Requirements

An APR system must satisfy Correctness, as well as the following security requirements: *Anonymity of Ratings under Full Corruption*, which formalises the strongest anonymity that can be achieved when the Rewarder and Reputation Holder are corrupted; *Anonymity of Ratings under a Corrupt Reputation Holder*, which ensures that ratings cannot be de-anonymised or linked by the Reputation Holder[6]; *Traceability*, which ensures that multiple ratings cannot be given on the same user per round; *Non–Frameability*, which ensures that users cannot be impersonated when giving ratings or claiming incentives; *Unforgeability of Reputation*, which ensures that a user cannot lie about their reputation, and *Fair Rewards*, which ensures that users can only successfully claim for the number of incentives they were awarded. We focus here on the Anonymity of Ratings and Fair Rewards requirements as these are the most novel, directly relating to the problem of incentivising anonymous ratings. However, the Traceability, Non–Frameability and Unforgeability of Reputation requirements are given in the full version of this paper [26].

We provide definitions in the computational model of cryptography. These are typically formulated as experiments in which an adversary, having access to a certain number of *oracles*, is challenged to produce an output. Such output captures an instance of the system in which the security requirement does not hold. In Figure 2, we provide the oracles used in our security requirements: `AddU`, `SndToU`, `SndToRH`, `AllocateRep`, `USK`, `Rate`, `TD`, `Incent`, `Collect`, based on notation from [6]. We give a high level description below:

[6] The case of a corrupt Rewarder is captured in the Anonymity of Ratings under Full Corruption requirement.

- `AddU` (Add User): creates an honest user *uid*.
- `SndToU` (Send to User): creates honest users when the adversary has corrupted the Reputation Holder. The adversary impersonates the RH, and engages in a $<$ Join, Issue $>$ protocol with an honest user.
- `SndToRH` (Send to RH): creates corrupted users, when the adversary has not corrupted the Reputation Holder. The adversary impersonates a user and engages in a $<$ Join, Issue $>$ protocol with an honest RH.
- `AllocateRep`: allows an adversary to obtain outputs of AllocateRep.
- `USK`: allows an adversary to obtain the secret key of an honest user.
- `Rate`: allows an adversary to perform Rate on behalf of an honest user.
- `TD`: allows an adversary to obtain a trapdoor associated to a rating that has been obtained through the `Rate` oracle.
- `Incent`: allows an adversary to obtain outputs of Incent.
- `Collect`: allows an adversary to obtain outputs of CollectIncent for a rating that has been output by the `Rate` oracle and then input to the `Incent` oracle.

All oracles have access to the following records maintained as global state which are initially set to $\emptyset$:

HL List of *uid*s of honest users. New honest users can be added by queries to the `AddU` oracle (for an honest RH) or `SndToU` oracle (for a corrupt RH).
CL List of corrupt users that have requested to join the group. New corrupt users can be added through the `SndToRH` oracle if the RH is honest. If the RH is corrupt, we do not keep track of corrupt users.
AL List of all queries to the `AllocateRep` oracle for corrupt users.
SL List of queries and outputs from the `Rate` oracle.
TDL List of queries to the `TD` oracle.
IL List of queries, and outputs of the `Incent` oracle.
CLL List of queries, and outputs of the `Collect` oracle.

Correctness An APR system is correct, if when Rate is input an honestly generated secret key and a reputation token, it will output a valid rating. Provided all ratings input to FormRep1 originate from different users it will output the correct function. Also, if Incent and CollectIncent are performed honestly on k valid ratings, the resulting incentive claims will be valid. Provided each incentive is only claimed once, FormRep2 will output $f_1(k)$. We give the full requirement in the full version of this paper [26].

Anonymity of Ratings We now give the requirements for both corruption settings that ensure ratings cannot be de-anonymised or linked by user, provided multiple ratings on the same user per round are not given. We also must ensure that ratings cannot be linked to the corresponding incentive claim. This is crucial to ensuring ratings are anonymous, as incentive claims are sent fully authenticated and so, if linkable to the corresponding rating, they could be used to de-anonymise such ratings.

Fig. 2. Oracles used in our Security Requirements

Anonymity of Ratings under Full Corruption. We first formally define anonymity of ratings in the case both the Rewarder and the Reputation Holder have been corrupted. In this setting, the following attack can always be mounted: The adversary, having corrupted the Rewarder and Reputation Holder, wishes to de-anonymise a specific rating and so simply only rewards this rating. The author of the rating then claims their reward from the Reputation Holder, revealing their identity. Such an attack is unavoidable when incentivising accurate feedback.

However, we can still provide some guarantee of anonymity, namely that the adversary should learn no more than the following: a user that has been rewarded n times per round is responsible for n of the rewarded ratings for that round. When $n = 1$ the above attack still holds, but this dishonest behaviour of the Rewarder can be detected as only one incentive would be publicly broadcast. Our security requirement achieves this by allowing the challenge rating to be input to the `Collect` oracle, on the condition that an additional rating authored by the other challenged user is added to the inputs. By including ratings originating from both challenged users, the incentives claimed by both of these users will increase by 1, and so the adversary cannot use this to trivially win. We note

that this notion implies the anonymity requirement when just the Rewarder is corrupted, i.e., it is the strongest of the two requirements.

In the security game the Reputation Holder and Rewarder are corrupted, and so the adversary can create corrupted users. The adversary chooses two honest users, as well as a feedback, a user who is the subject of the feedback, and a reputation. The adversary must give reputation tokens for each user for this reputation. The adversary is returned with a challenge rating authored by one of these users, with this reputation, on this feedback and user (subject), and they must guess which user authored the rating. The challenge rating as well as another rating authored by the other challenged user is saved in RL, for later use in the `Collect` oracle. The adversary can create honest users with the `SndToU` oracle and obtain their ratings with the `Rate` oracle. However they cannot query to the `Rate` oracle either of the users that were challenged as well as the challenge subject/round. Otherwise the $\mathsf{FormRep1}$ algorithm could be used to trivially win, due to the detection of multiple ratings on the same user/round. We also must check that both ratings computed from the challenged users are valid, to ensure that both ω_0 or ω_1 output by the adversary were correctly formed. The adversary can also reveal the trapdoor from each `Rate` oracle query with the `TD` oracle, but not for the challenge ratings as this would lead to a trivial win by detecting double claims with $\mathsf{FormRep2}$. They also have access to an `Incent` oracle. The adversary can query incentives from the `Incent` oracle, that originate from the `Rate` oracle, to the `Collect` oracle. If they include the challenge rating, an additional rating from the other challenged user is added to the inputs. The adversary is returned with the incentive claims for these ratings along with the user who claims them. This captures the fact that claiming incentives should not violate the anonymity of ratings. We give the full game below:

Experiment: $\mathsf{Exp}^{\text{anon-fullcorr}}_{\mathcal{A},\mathsf{APR}}(\tau, f_1, f_2)$

$b \leftarrow\!\!\$ \{0,1\}, \mathsf{RL}, \mathsf{RL}^{\dagger} \leftarrow \emptyset, \mathsf{param} \leftarrow\!\!\$ \mathsf{Setup}(1^{\tau}, f_1, f_2), (rhsk, rhpk) \leftarrow\!\!\$ \mathsf{RHKeyGen}(\mathsf{param})$

$(rwsk, rwpk) \leftarrow\!\!\$ \mathsf{RWKeyGen}(\mathsf{param}), gpk \leftarrow (\mathsf{param}, rwpk, rhpk)$

$(\mathrm{st}, uid_0^*, uid_1^*, l^*, fb^*, uid'^*, r^*, \omega_0, \omega_1) \leftarrow\!\!\$ \mathcal{A}^{\mathtt{SndToU},\mathtt{Rate},\mathtt{TD},\mathtt{Incent},\mathtt{Collect}}(\mathsf{choose}, gpk, rhsk, rwsk)$

if $uid_0^*, uid_1^* \notin \mathsf{HL}$ or $\mathbf{gsk}[uid_0^*], \mathbf{gsk}[uid_1^*] = \perp$ **return** $\perp$

$\forall b' \in \{0,1\} \quad (td_{b'}^*, \rho_{b'}^*) \leftarrow\!\!\$ \mathsf{Rate}(\mathbf{gsk}[uid_{b'}^*], gpk, fb^*, uid'^*, l^*, r^*, \omega_{b'})$

// Compute both ratings for use in `Collect` oracle and to check ω_0, ω_1

$\mathsf{RL} \leftarrow \{(uid_{b'}^*, uid'^*, fb^*, r^*, \rho_{b'}^*, td_{b'}^*, l^*) : b' \in \{0,1\}\}$ // Save both ratings for use in `Collect`

$d \leftarrow\!\!\$ \mathcal{A}^{\mathtt{SndToU},\mathtt{Rate},\mathtt{TD},\mathtt{Incent},\mathtt{Collect}}(\mathsf{guess}, \mathrm{st}, \rho_b^*)$

if ρ_0^* or $\rho_1^* = \perp$ or $\exists b' \in \{0,1\}$ s.t $(uid_{b'}^*, uid'^*, \cdot, \cdot, \cdot, \cdot, l^*) \in \mathsf{SL}$

// Check ω_0, ω_1 are both valid and $\mathsf{FormRep1}$ can't be used to trivially win by detecting multiple ratings

return $d \leftarrow\!\!\$ \{0,1\}$

if $d = b$ **return** 1 **else return** 0

An APR system satisfies Anonymity of Ratings under Full Corruption if for all functions f_1, f_2, for all polynomial time adversaries $\mathcal{A}$, the following advantage is negligible in τ:

$$|\Pr[\mathsf{Exp}^{anon-fullcorr}_{\mathcal{A},\mathsf{APR}}(\tau, f_1, f_2) = 1] - 1/2|.$$

Anonymity of Ratings under a Corrupt Reputation Holder. We next define anonymity in the setting where the Reputation Holder has been corrupted, but not the Rewarder. This means that the adversary now does not know which ratings have been rewarded. The challenge rating and a rating authored by the other challenged user are now stored in list $\mathsf{RL}^{\dagger}$. The adversary has full access to the `Collect` oracle, modelling the role of the Reputation Holder. However, if the challenge rating is input to the `Incent` oracle, the rating authored by the other challenged user stored in $\mathsf{RL}^{\dagger}$ is also added to the inputs. The `Incent` oracle shuffles the outputs. This represents that the Reputation Holder no longer knows which rating is linked to each incentive.

Experiment: $\mathsf{Exp}^{\text{anon-rh}}_{\mathcal{A},\mathsf{APR}}(\tau, f_1, f_2)$

$b \leftarrow\!\!\$ \{0,1\}, \mathsf{RL}, \mathsf{RL}^{\dagger} \leftarrow \emptyset, \mathsf{param} \leftarrow\!\!\$ \mathsf{Setup}(1^{\tau}, f_1, f_2), (rhsk, rhpk) \leftarrow\!\!\$ \mathsf{RHKeyGen}(\mathsf{param})$

$(rwsk, rwpk) \leftarrow\!\!\$ \mathsf{RWKeyGen}(\mathsf{param}), gpk \leftarrow (\mathsf{param}, rwpk, rhpk)$

$(\mathrm{st}, uid_0^*, uid_1^*, l^*, fb^*, uid'^*, r^*, \omega_0, \omega_1) \leftarrow\!\!\$ \mathcal{A}^{\mathtt{SndToU},\mathtt{Rate},\mathtt{TD},\mathtt{Incent},\mathtt{Collect}}(\mathsf{choose}, gpk, rhsk)$

if $uid_0^*, uid_1^* \notin \mathsf{HL}$ or $\mathbf{gsk}[uid_0^*], \mathbf{gsk}[uid_1^*] = \perp$ **return** $\perp$

$\forall b' \in \{0,1\} \quad (td_{b'}^*, \rho_{b'}^*) \leftarrow\!\!\$ \mathsf{Rate}(\mathbf{gsk}[uid_{b'}^*], gpk, fb^*, uid'^*, l^*, r^*, \omega_{b'})$

// Compute both ratings for use in Incent oracle and to check ω_0, ω_1

$\mathsf{RL}^{\dagger} \leftarrow \{(uid_{b'}^*, uid'^*, fb^*, r^*, \rho_{b'}^*, td_{b'}^*, l^*) : b' \in \{0,1\}\}$ // Save both ratings for use in Incent

$d \leftarrow\!\!\$ \mathcal{A}^{\mathtt{SndToU},\mathtt{Rate},\mathtt{TD},\mathtt{Incent},\mathtt{Collect}}(\mathsf{guess}, \mathrm{st}, \rho_b^*)$

if ρ_0^* or $\rho_1^* = \perp$ or $\exists b' \in \{0,1\}$ s.t $(uid_{b'}^*, uid'^*, \cdot, \cdot, \cdot, \cdot, l^*) \in \mathsf{SL}$

// Check ω_0, ω_1 are both valid and FormRep1 can't be used to trivially win by detecting multiple ratings

return $d \leftarrow\!\!\$ \{0,1\}$

if $d = b$ **return** 1 **else return** 0

An APR system satisfies Anonymity of Ratings under a Corrupt Reputation Holder if for all f_1, f_2, for all polynomial time adversaries $\mathcal{A}$, the following advantage is negligible in τ:

$$|\Pr[\mathsf{Exp}^{anon-rh}_{\mathcal{A},\mathsf{APR}}(\tau, f_1, f_2) = 1] - 1/2|.$$

Fair Rewards This requirement ensures that an adversary cannot increase the number of incentives they were allocated, or steal incentives allocated to other users. In the security game the Rewarder and the Reputation Holder are corrupted, so the adversary can create corrupted users. The adversary is given the `SndToU` and `Rate` oracles to create honest users, and obtain their ratings. They have access to the `Collect` oracles to obtain incentive claims on incentives obtained from the Rate oracle followed by the `Incent` oracle. They have access to the trapdoor oracle, to obtain trapdoors associated to ratings output by `Rate`. The adversary must choose k_1 incentives obtained from the `Incent` oracle, and k_2 valid incentive claims, not output by the `Collect` oracle, corresponding to a single user identifier. If FormRep2 doesn't detect cheating, and more incentive claims are output than incentives corresponding to ratings not obtained through the `Rate` oracle or queried to the trapdoor oracle, then the adversary wins. We give the full game below:

Experiment: $\mathsf{Exp}^{fair-rew}_{\mathcal{A},\mathsf{APR}}(\tau, f_1, f_2)$

$\mathsf{RL}, \mathsf{RL}^{\dagger} \leftarrow \emptyset, \mathsf{param} \leftarrow_{\$} \mathsf{Setup}(1^{\tau}, f_1, f_2), (rhsk, rhpk) \leftarrow_{\$} \mathsf{RHKeyGen}(\mathsf{param})$

$(rwsk, rwpk) \leftarrow_{\$} \mathsf{RWKeyGen}(\mathsf{param}), gpk \leftarrow (\mathsf{param}, rwpk, rhpk)$

$(uid, (\sigma_1, \cdots \sigma_{k_2}), (t_1, \cdots t_{k_1}), l) \leftarrow_{\$} \mathcal{A}^{\mathtt{SndToU},\mathtt{Rate},\mathtt{TD},\mathtt{Incent},\mathtt{Collect}}(gpk, rwsk, rhsk)$

if $\exists i \in [k_1]$ s.t $(t_i, (fb_i, uid'_i, r_i, \rho_i)) \notin \mathsf{IL}$ **return** 0

return 1 **if** the following conditions hold

$\forall i \in [k_2]$ σ_i not returned by `Collect` oracle and $\mathsf{VerifyIncent}(uid, \sigma_i, (t_1, \cdots, t_{k_1})) = 1$ and

$\mathsf{FormRep2}(uid, \sigma_1, \cdots \sigma_{k_2}, t_1, \cdots t_{k_1}, gpk) \neq \perp$ and

$k_2 > |\{i \in [k_1] : (\cdot, uid'_i, fb_i, r_i, \rho_i, \cdot, l) \notin \mathsf{SL} \text{ or } (fb_i, uid'_i, l, r_i, \rho_i) \in \mathsf{TDL}\}|$

An APR system satisfies Fair Rewards if for all functions f_1, f_2, for all polynomial time adversaries $\mathcal{A}$, the advantage $\Pr[\mathsf{Exp}^{fair-rew}_{\mathcal{A},\mathsf{APR}}(\tau, f_1, f_2) = 1]$ is negligible in τ.

3 Construction

We propose a construction for an APR system which makes use of three building blocks: Linkable Ring Signatures (LRS), a modified Direct Anonymous Attestation (DAA*) scheme and a public–key encryption scheme.

Ring signatures [46] allow users to sign on behalf of a ring of users, without revealing their identity within the ring. There is no central entity involved, and users generate their own signing and verification keys. Linkable ring signatures [36] allow for the public linking of signatures by signer. We exploit these features to allow for incentivising accurate ratings as follows. Each rating includes a freshly generated verification key encrypted under the public key of the Rewarder, and the user who has generated the rating stores the corresponding signing key as a trapdoor. The Rewarder publishes these decrypted verification keys as incentives. Then to claim an incentive the user uses the signing key to sign a ring signature on their user identifier with respect to the ring of verification keys given as incentives. The anonymity of Linkable Ring Signatures ensures that claiming incentives will not de-anonymise ratings. The unforgeability property ensures that only users that have been rewarded can claim an incentive, and the linking function ensures that only one reward can be claimed per rating.

Direct Anonymous Attestation (DAA) [13] allows users to sign on behalf of a group, whilst remaining anonymous within the group. The user-controlled linkability feature, where two signatures on the same basename by the same user are linked, whilst all other signatures are unlinkable, can be used to detect multiple feedback on the same subject. In our setting, the basename can be set to be the user who is the subject of the feedback and the round. In our system we also wish to ensure feedback is weighted by reputation. However, this must also be balanced with anonymity of feedback. For this to be possible the reputation of users must be coarse-grained enough that they cannot be identified by their reputation. To ensure this, we bind reputation into a Direct Anonymous Attestation scheme, which we will call a DAA* scheme. Now a user proves their reputation when signing, allowing for the weighting of feedback.

3.1 Public-Key Encryption Schemes

Our scheme makes use of a public–key encryption scheme, which consists of the following: $\mathsf{EncSetup}(1^\tau)$, which is input the security parameter 1^τ and outputs parameters $\mathsf{param}_{\mathsf{Enc}}$; $\mathsf{EncKeyGen}(\mathsf{param}_{\mathsf{Enc}})$, which is input the parameters and outputs secret key sk and the public key pk; $\mathsf{Enc}(pk, m)$, which is input the public key pk and a message m from the message space, and outputs a ciphertext c; and $\mathsf{Dec}(sk, c)$, which is input the secret key sk and a ciphertext c, and outputs a message m or a decryption failure $\perp$. We require the encryption scheme to be correct and satisfy indistinguishability under adaptive chosen ciphertext attacks.

3.2 Linkable Ring Signatures

We use the model in [4] for one-time linkable ring signatures, which gives the strongest security yet. The scheme from [4] has the shortest signatures to date. We give the security requirements: Correctness Linkability, Linkable Anonymity, Non–Frameability and Unforgeability in the full version [26].

Definition 1 (Linkable Ring Signatures.)**.** A linkable ring signature scheme LRS is given by polynomial time algorithms $(\mathsf{LRKeyGen}, \mathsf{LRSign}, \mathsf{LRVerify}, \mathsf{LRLink})$:

$\mathsf{LRKeyGen}(1^\tau)$: takes as input the security parameter 1^τ and outputs a pair (vk, sk) of verification and signing keys.

$\mathsf{LRSign}(sk, m, R)$: takes as input a signing key sk, a message m, and a list of verification keys $R = (vk_1, ..., vk_q)$, and outputs a signature Σ.

$\mathsf{LRVerify}(R, m, \Sigma)$: takes as input a ring $R = (vk_1, ..., vk_q)$, a message m, and a signature Σ, and outputs either 0 or 1.

$\mathsf{LRLink}(\Sigma_1, \Sigma_2, m_1, m_2)$: is input two signatures/ messages, outputs 0 or 1.

3.3 DAA* Signatures

The security model of DAA* closely follows that of pre–DAA signatures [7]. We give the security requirements for DAA* signatures in the full version [26].

Definition 2 (DAA*.)**.** A DAA* scheme consists of the following algorithms:

$\mathsf{DAA*Setup}(1^\tau)$: input the security parameter τ, outputs parameters param.

$\mathsf{DAA*KeyGen}(\mathsf{param})$: input the parameters param, outputs the group public key gpk, and the issuing secret key isk.

$\langle \mathsf{DAA*Join}(gpk), \mathsf{DAA*Issue}(isk, gpk) \rangle$: a user i joins the group by engaging in an interactive protocol with the Issuer. The user i and Issuer perform algorithms DAA*Join and DAA*Issue respectively. These are input a state and an incoming message respectively, and output an updated state, an outgoing message, and a decision, either cont, accept, or reject. The initial input to DAA*Join is the group public key, whereas the initial input to DAA*Issue is the issuer secret key, isk, and the group public key. If the issuer accepts, DAA*Join has a private output of $\mathbf{gsk}[i]$, DAA*Issue has a private output of $\mathbf{reg}[i]$.

$\mathsf{DAA*Update}(r, t, isk, i, \mathbf{reg}, gpk)$: input a reputation r, a time t, the issuing secret key isk, a user i, the registration list $\mathbf{reg}$, gpk. Outputs a token ω.

DAA*Sign$(bsn, m, \mathbf{gsk}[i], \omega, gpk, r, t)$: input a basename bsn, a message m, a user secret key $\mathbf{gsk}[i]$, a token ω output by DAA*Update, a group public key gpk, a reputation r and time t. It checks that ω is valid for user i, reputation r and time t and outputs a signature Ω. Otherwise it outputs $\perp$.

DAA*Verify$(bsn, m, r, t, \Omega, gpk)$: input a basename bsn, a message m, a reputation r, time t, a signature Ω, and a group public key gpk. It outputs 1 if Ω is valid for the item I, reputation r and time t, and 0 otherwise.

DAA*Link$((bsn_0, m_0, r_0, t_0, \Omega_0), (bsn_1, m_1, r_1, t_1, \Omega_1), gpk)$: input two signatures Ω_0, Ω_1 each on a basename, a message, a reputation, a time, and a group public key gpk. It outputs 1 if both signatures are valid, $bsn_0 = bsn_1$ and the two signatures have the same author, and 0 otherwise.

DAA*Identify$_T(\mathcal{T}, gsk)$: outputs 1 if $\mathcal{T}$ corresponds to a valid transcript of $<$ DAA*Join, DAA*Issue $>$, with output gsk to DAA*Join, and otherwise 0.

DAA*Identify$_S(bsn, m, r, t, \Omega, gsk)$: outputs 1 if the signature Ω could have been produced with user secret key gsk, and 0 otherwise.

3.4 Our Construction

We now present our construction that securely realizes an APR system, using a PKE scheme, an LRS scheme and a DAA* scheme. We give our construction in Figure 3, except for the $<$ Join, Issue $>$ protocol which is identical to the $<$ DAA*Join, DAA*Issue $>$ protocol for DAA* signatures such that DAA*Join is input $rhpk$, and DAA*Issue is input $(rhsk, rhpk)$.

3.5 Security of Our Construction

We show that our construction satisfies the security requirements for an APR system defined in Section 2. We need one further property than the security of the LRS and DAA* building blocks. In the $<$ Join, Issue $>$ protocol, the RH must be sent an SPK of the user's secret key. SPK denotes a signature proof of knowledge, that is a non-interactive transformation of a proof PK. These proofs can be extended to be online-extractable [24], by verifiably encrypting the witness to a public key defined in the common reference string. We require the proof system to be simulation-sound, online–extractable and zero-knowledge. We give further details on the proof protocols used in the full version [26].

Theorem 1. *The construction presented in Figure 3 is a secure APR, as defined in Section 2, if the LRS scheme, DAA* scheme and PKE scheme used are secure, and the SPK is online extractable, simulation sound, and zero-knowledge.*

The detailed proofs of Lemmata 1-6 are given in the full version [26].

Lemma 1. *The construction satisfies Anonymity of Ratings under Full–Corruption if the LRS and DAA* schemes satisfy Anonymity, and the SPK is zero-knowledge.*

Proof intuition. A distinguisher between the original game and one where the challenged user identifiers are swapped in the `Collect` oracle when the challenge rating is input, can break the anonymity of linkable ring signatures. A reduction can now be made to the anonymity of our DAA* scheme. □

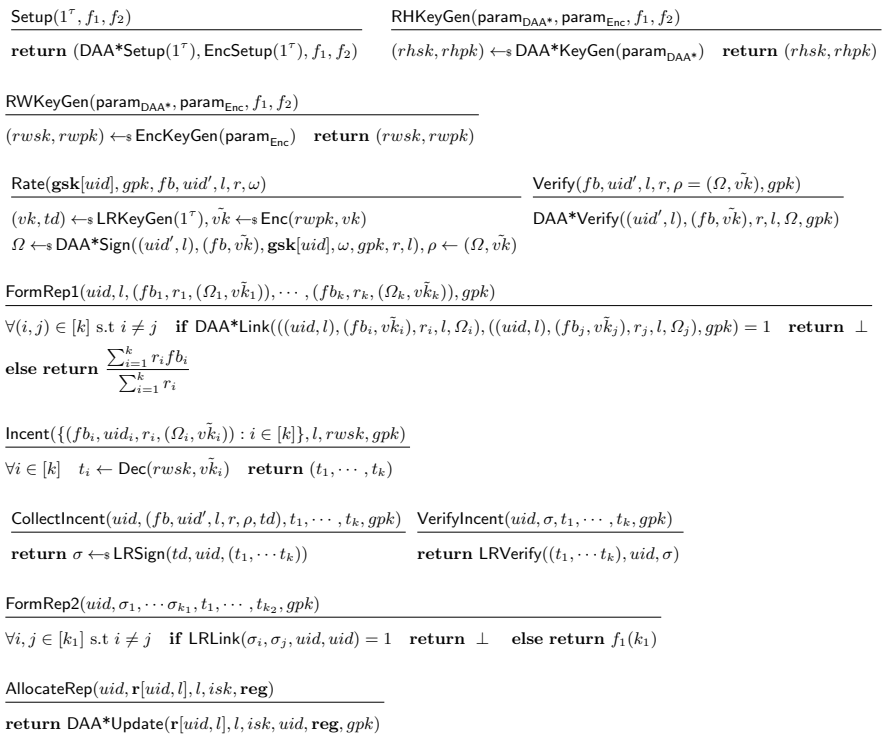

Fig. 3. Our APR construction

Lemma 2. *The construction satisfies Anonymity of Ratings under a Corrupt Reputation Holder if the DAA* scheme satisfies Anonymity and the PKE scheme satisfies indistinguishability under adaptive chosen ciphertext attacks, and the SPK is zero-knowledge.*

Proof intuition. A distinguisher between the original game and a game where the `Collect` oracle, on input an incentive from the ratings in $\mathsf{RL}^\dagger$, swaps the user identifiers, can break the IND–CCA2 security of the encryption scheme. A reduction can now be made to the anonymity of our DAA* scheme. □

Lemma 3. *The construction satisfies Traceability if the DAA* scheme satisfies both Traceability and Non–Frameability, and the SPK is online extractable and simulation sound.*

Lemma 4. *The construction satisfies Non–Frameability if the LRS and DAA* schemes both satisfy Non–Frameability, and the SPK is zero-knowledge.*

Lemma 5. *The construction satisfies Unforgeability of Reputation if the DAA* scheme satisfies Unforgeability of Reputation, and the SPK is online extractable and simulation sound.*

Lemma 6. *The construction satisfies Fair Rewards if the LRS scheme satisfies Linkability and Non–Frameability.*

Proof intuition. An adversary breaks fair rewards either by "stealing" an incentive from an honest user, in which case we could break the non–frameability of LRS, or by expanding the incentives that were fairly allocated to corrupted users, in which case we could break the Linkability of LRS. □

3.6 Concrete Instantiation and Efficiency

We give a DAA* construction, and prove that it securely realizes a DAA* scheme in the full version [26], assuming the LRSW [37], DCR [41], and DDH assumptions and the random oracle model. The $<$ DAA*Join, DAA*Issue $>$ protocol already contains a suitable SPK of the user secret key. A linkable ring signature scheme that securely realises the model in Section 3.2 is given in [4]. An incentive claim would have size $\log(l)\mathsf{poly}(\tau)$, where l is the number of incentives. This is the current state of the art for linkable ring signatures, and is reasonable, albeit large. Ratings are reasonably small, and consist of 7 τ-bit elements, and an encryption of 3 commitments.

4 Conclusion and Future Work

We give a security model for an anonymous peer rating system APR that allows accurate ratings to be incentivised, feedback to be weighted by reputation, and multiple feedback on the same subject to be detected, whilst still ensuring ratings remain anonymous. We use Linkable Ring Signatures and a modification of DAA to build a construction that is secure under these requirements.

The DAA and Linkable Ring Signature primitives are not inherent in realising our anonymous peer ratings system. Different primitives could be used to build constructions that are more efficient or rely on different assumptions.

In a peer rating system, a high reputation score leads to a real payoff for users, corresponding to an increase in utility. When increasing one's utility is the ultimate goal, game theory helps to gain new insights. A peer rating system formalised through game theory, which also follows the strategies of weighting feedback and incentivising accurate ratings, is proposed in [50] and experimentally simulated when used in collaborative intrusion detection systems in [20]. It is shown in [50] to what extent it pays off for users to rate accurately given the size of incentives and the size of the coalition(s) of dishonest users. However, anonymity of ratings is not taken into account and a fully trusted central authority receives the ratings and issues the incentives. As future work, we want to determine game theoretically whether our scheme incentivises accurate ratings.

References

1. Gnutella. `https://en.wikipedia.org/wiki/Gnutella`, accessed 30 August, 2019.
2. Abdalla, M., Bellare, M., Catalano, D., Kiltz, E., Kohno, T., Lange, T., Malone-Lee, J., Neven, G., Paillier, P., Shi, H.: Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 205–222. Springer, Heidelberg (Aug 2005)

3. Androulaki, E., Choi, S.G., Bellovin, S.M., Malkin, T.: Reputation systems for anonymous networks. In: International Symposium on Privacy Enhancing Technologies Symposium. pp. 202–218. Springer (2008)
4. Backes, M., Döttling, N., Hanzlik, L., Kluczniak, K., Schneider, J.: Ring signatures: Logarithmic-size, no setup - from standard assumptions. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019, Part III. LNCS, vol. 11478, pp. 281–311. Springer, Heidelberg (May 2019)
5. Bellare, M., Boldyreva, A., Desai, A., Pointcheval, D.: Key-privacy in public-key encryption. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 566–582. Springer, Heidelberg (Dec 2001)
6. Bellare, M., Shi, H., Zhang, C.: Foundations of group signatures: The case of dynamic groups. In: Menezes, A. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 136–153. Springer, Heidelberg (Feb 2005)
7. Bernhard, D., Fuchsbauer, G., Ghadafi, E.M., Smart, N.P., Warinschi, B.: Anonymous attestation with user-controlled linkability. Int. J. Inf. Secur. 12(3), 219–249 (Jun 2013)
8. Bethencourt, J., Shi, E., Song, D.: Signatures of reputation. In: Sion, R. (ed.) FC 2010. LNCS, vol. 6052, pp. 400–407. Springer, Heidelberg (Jan 2010)
9. Blömer, J., Bobolz, J., Diemert, D., Eidens, F.: Updatable anonymous credentials and applications to incentive systems. In: Cavallaro, L., Kinder, J., Wang, X., Katz, J. (eds.) ACM CCS 2019. pp. 1671–1685. ACM Press (Nov 2019)
10. Blömer, J., Eidens, F., Juhnke, J.: Practical, anonymous, and publicly linkable universally-composable reputation systems. In: Smart, N.P. (ed.) CT-RSA 2018. LNCS, vol. 10808, pp. 470–490. Springer, Heidelberg (Apr 2018)
11. Blömer, J., Juhnke, J., Kolb, C.: Anonymous and publicly linkable reputation systems. In: Böhme, R., Okamoto, T. (eds.) FC 2015. LNCS, vol. 8975, pp. 478–488. Springer, Heidelberg (Jan 2015)
12. Bobolz, J., Eidens, F., Krenn, S., Slamanig, D., Striecks, C.: Privacy-preserving incentive systems with highly efficient point-collection. In: To apppear at Proceedings of the 2020 ACM Asia Conference on Computer and Communications Security (2020)
13. Brickell, E.F., Camenisch, J., Chen, L.: Direct anonymous attestation. In: Atluri, V., Pfitzmann, B., McDaniel, P. (eds.) ACM CCS 2004. pp. 132–145. ACM Press (Oct 2004)
14. Brinckman, A., Deelman, E., Gupta, S., Nabrzyski, J., Park, S., Ferreira da Silva, R., Taylor, I.J., Vahi, K.: Collaborative circuit designs using the CRAFT repository. Future Generation Computer Systems 94, 841–853 (2019)
15. Camenisch, J., Kohlweiss, M., Rial, A., Sheedy, C.: Blind and anonymous identity-based encryption and authorised private searches on public key encrypted data. In: Jarecki, S., Tsudik, G. (eds.) PKC 2009. LNCS, vol. 5443, pp. 196–214. Springer, Heidelberg (Mar 2009)
16. Chaum, D., van Heyst, E.: Group signatures. In: Davies, D.W. (ed.) EUROCRYPT'91. LNCS, vol. 547, pp. 257–265. Springer, Heidelberg (Apr 1991)
17. Chen, L., Li, Q., Martin, K.M., Ng, S.L.: A privacy-aware reputation-based announcement scheme for vanets. In: Wireless Vehicular Communications (WiVeC), 2013 IEEE 5th International Symposium on. pp. 1–5. IEEE (2013)
18. Chen, L., Li, Q., Martin, K.M., Ng, S.L.: Private reputation retrieval in public - a privacy-aware announcement scheme for vanets. IET Information Security 11(4), 204–210 (July 2017)

19. Chuang, J.: Designing incentive mechanisms for peer-to-peer systems. In: 1st IEEE International Workshop on Grid Economics and Business Models, 2004. GECON 2004. pp. 67–81. IEEE (2004)
20. Cordero, C.G., Traverso, G., Nojoumian, M., Habib, S.M., Mühlhäuser, M., Buchmann, J.A., Vasilomanolakis, E.: Sphinx: a colluder-resistant trust mechanism for collaborative intrusion detection. IEEE Access 6, 72427–72438 (2018)
21. Dellarocas, C.: Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior. Proceedings of the 2nd ACM conference on Electronic commerce (Mar 2001)
22. Douceur, J.R.: The sybil attack. In: Druschel, P., Kaashoek, F., Rowstron, A. (eds.) Peer-to-Peer Systems. pp. 251–260. Springer Berlin Heidelberg, Berlin, Heidelberg (2002)
23. El Kaafarani, A., Katsumata, S., Solomon, R.: Anonymous reputation systems achieving full dynamicity from lattices. In: Proceedings of the 22nd International Conference on Financial Cryptography and Data Security (FC) (2018)
24. Fischlin, M.: Communication-efficient non-interactive proofs of knowledge with online extractors. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 152–168. Springer, Heidelberg (Aug 2005)
25. Garms, L., Martin, K.M., Ng, S.L.: Reputation schemes for pervasive social networks with anonymity. In: Proceedings of the fifteenth International Conference on Privacy, Security and Trust (PST 2017). pp. 1–6. IEEE (Aug 2017)
26. Garms, L., Quaglia, E., Ng, S.L., Traverso, G.: Anonymity and rewards in peer rating systems. Cryptology ePrint Archive, Report 2020/790 (2020), `https://eprint.iacr.org/2020/790`
27. Garms, L., Quaglia, E.A.: A new approach to modelling centralised reputation systems. In: Buchmann, J., Nitaj, A., eddine Rachidi, T. (eds.) AFRICACRYPT 19. LNCS, vol. 11627, pp. 429–447. Springer, Heidelberg (Jul 2019)
28. Giannoulis, M., Kondylakis, H., Marakakis, E.: Designing and implementing a collaborative health knowledge system. Expert Systems with Applications 126, 277–294 (2019)
29. Hartung, G., Hoffmann, M., Nagel, M., Rupp, A.: BBA+: Improving the security and applicability of privacy-preserving point collection. In: Thuraisingham, B.M., Evans, D., Malkin, T., Xu, D. (eds.) ACM CCS 2017. pp. 1925–1942. ACM Press (Oct / Nov 2017)
30. Hawley, M., Howard, P., Koelle, R., Saxton, P.: Collaborative security management: Developing ideas in security management for air traffic control. In: 2013 International Conference on Availability, Reliability and Security. pp. 802–806 (Sep 2013)
31. Hoffman, K., Zage, D., Nita-Rotaru, C.: A survey of attack and defense techniques for reputation systems. ACM Computing Surveys 42(1), 1:1–1:31 (Dec 2009)
32. Hopwood, D., Bowe, S., Hornby, T., Wilcox, N.: Zcash protocol specification. Tech. rep., Zerocoin Electric Coin Company (2016)
33. Jager, T., Rupp, A.: Black-box accumulation: Collecting incentives in a privacy-preserving way. PoPETs 2016(3), 62–82 (Jul 2016)
34. Jøsang, A., Golbeck, J.: Challenges for robust trust and reputation systems. In: 5th International Workshop on Security and Trust Management (STM 2009) (2009)
35. Libert, B., Paterson, K.G., Quaglia, E.A.: Anonymous broadcast encryption: Adaptive security and efficient constructions in the standard model. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) PKC 2012. LNCS, vol. 7293, pp. 206–224. Springer, Heidelberg (May 2012)

36. Liu, J.K., Wei, V.K., Wong, D.S.: Linkable spontaneous anonymous group signature for ad hoc groups (extended abstract). In: Wang, H., Pieprzyk, J., Varadharajan, V. (eds.) ACISP 04. LNCS, vol. 3108, pp. 325–335. Springer, Heidelberg (Jul 2004)
37. Lysyanskaya, A., Rivest, R.L., Sahai, A., Wolf, S.: Pseudonym systems. In: Heys, H.M., Adams, C.M. (eds.) SAC 1999. LNCS, vol. 1758, pp. 184–199. Springer, Heidelberg (Aug 1999)
38. Mármol, F.G., Pérez, G.M.: Security threats scenarios in trust and reputation models for distributed systems. Computers & Security 28(7), 545–556 (2009)
39. Milutinovic, M., Dacosta, I., Put, A., Decker, B.D.: uCentive: An efficient, anonymous and unlinkable incentives scheme. In: 2015 IEEE Trustcom/BigDataSE/ISPA. vol. 1, pp. 588–595 (2015)
40. Nabuco, O., Bonacin, R., Fugini, M., Martoglia, R.: Web2touch 2016: Evolution and security of collaborative web knowledge. In: 2016 IEEE 25th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE). pp. 214–216 (June 2016)
41. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT'99. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (May 1999)
42. Papaioannou, T.G., Stamoulis, G.D.: An incentives' mechanism promoting truthful feedback in peer-to-peer systems. In: CCGrid 2005. IEEE International Symposium on Cluster Computing and the Grid, 2005. vol. 1, pp. 275–283 (May 2005)
43. Pavlov, E., Rosenschein, J.S., Topol, Z.: Supporting privacy in decentralized additive reputation systems. In: International Conference on Trust Management. pp. 108–119. Springer (2004)
44. Petrlic, R., Lutters, S., Sorge, C.: Privacy-preserving reputation management. In: Proceedings of the 29th Annual ACM Symposium on Applied Computing. pp. 1712–1718. SAC '14, ACM, New York, NY, USA (2014)
45. Pingel, F., Steinbrecher, S.: Multilateral secure cross-community reputation systems for internet communities. In: International Conference on Trust, Privacy and Security in Digital Business. pp. 69–78. Springer (2008)
46. Rivest, R.L., Shamir, A., Tauman, Y.: How to leak a secret. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 552–565. Springer, Heidelberg (Dec 2001)
47. Schiffner, S., Clauß, S., Steinbrecher, S.: Privacy and liveliness for reputation systems. In: Martinelli, F., Preneel, B. (eds.) Public Key Infrastructures, Services and Applications. pp. 209–224. Springer Berlin Heidelberg (2010)
48. Sillaber, C., Sauerwein, C., Mussmann, A., Breu, R.: Data quality challenges and future research directions in threat intelligence sharing practice. In: Proceedings of the 2016 ACM on Workshop on Information Sharing and Collaborative Security. pp. 65–70. WISCS '16, ACM, New York, NY, USA (2016)
49. Sun, Y.L., Han, Z., Yu, W., Ray Liu, K.J.: Attacks on trust evaluation in distributed networks. In: 2006 40th Annual Conference on Information Sciences and Systems. pp. 1461–1466 (2006)
50. Traverso, G., Butin, D., Buchmann, J.A., Palesandro, A.: Coalition-resistant peer rating for long-term confidentiality. In: 2018 16th Annual Conference on Privacy, Security and Trust (PST). pp. 1–10 (Aug 2018)
51. Zhai, E., Wolinsky, D.I., Chen, R., Syta, E., Teng, C., Ford, B.: AnonRep: towards tracking-resistant anonymous reputation. In: 13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16). pp. 583–596. USENIX Association (2016)