De Cifris Trends
in
Cryptographic Protocols

University of Trento and *De Componendis Cifris*
November 2023

Lecture 9

Protocols for Peer Review Systems

Elizabeth A. Quaglia

Royal Holloway, University of London

Overview of this talk

In this talk we will

- introduce **Peer Rating Systems** (PRS) (models, definitions, security requirements)
- explore the importance and challenges of **anonymity** in PRS
- **define** anonymous PRS
- show a **construction** of anonymous PRS
- discuss **practical aspects** and **future work**

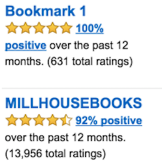
Introducing Peer Rating Systems

Intuitively, a *peer rating system* (PRS) assigns a user a **reputation value**, based on feedback from other users.
PRS are also known as reputation systems.

Introducing Peer Rating Systems

Intuitively, a *peer rating system* (PRS) assigns a user a **reputation value**, based on feedback from other users.
PRS are also known as reputation systems.

The reputation value is a metric of trustworthiness or merit.
It plays a fundamental role in our online activities, e.g., Uber, Amazon, AirBnb, Trip Advisor...



Bookmark 1
★★★★★ 100% positive over the past 12 months. (631 total ratings)

MILLHOUSEBOOKS
★★★★☆ 92% positive over the past 12 months. (13,956 total ratings)

Centralised vs Decentralised PRS

There are two models for PRS.

Centralised vs Decentralised PRS

There are two models for PRS.

Centralised PRS

In centralised PRS a **central reputation server** enrols users and forms reputation values on these users.

In this talk, we focus on this model since the reputation systems used by most service providers such as Airbnb, Uber and Amazon are of this type.

Centralised vs Decentralised PRS

There are two models for PRS.

Centralised PRS

In centralised PRS a **central reputation server** enrols users and forms reputation values on these users.
In this talk, we focus on this model since the reputation systems used by most service providers such as Airbnb, Uber and Amazon are of this type.

Decentralised PRS

Decentralised (or distributed) systems have **no** reputation server and use **local reputation values**, i.e., reputation values created by users on other users.
This means a user does not have a unique reputation value, but many other users hold their own reputation value for them.

Defining a PRS

Typically, a PRS is defined as follows [GQ19].

Defining a PRS

Typically, a PRS is defined as follows [GQ19].

Entities involved: A set of users $\{U_i\}$ and the central reputation server CS.

Defining a PRS

Typically, a PRS is defined as follows [GQ19].

Entities involved: A set of users $\{U_i\}$ and the central reputation server CS.

Defining algorithms:

- **Setup** Run by CS to setup the system's parameters and keys.
- **AllocateReputation** Run by CS, it produces tokens to allow users to prove their reputation.
- **PostItem** Run by U_i, on input an item and their last reputation it outputs a proof of enrolment and current reputation.
- **CheckItem** Run by U_j, it allows them to verify an item is valid.
- **SendFB** Run by U_j to give feedback on an item.
- **VerifyFB** Run by CS to check the feedback is valid.
- **LinkFB** Run by CS to check that there is no feedback by the same user on this item.
- **ReceiveFB** Run by CS to receive feedback on an item.

An example

Using a PRS for a **car pooling app**

- Whenever a driver wishes to update their reputation, they request the CS run AllocateReputation to obtain a token for their reputation. They are incentivised to do this by the fact the reputation is displayed alongside the time it was allocated.
- When they wish to give a ride, they use their most recent token to post an item with PostItem, which can be verified by passengers with CheckItem.
- The passenger can then pay using some anonymous payment system. After the ride, their passenger can then give feedback on this item to the CS using SendFB.
- The CS uses ReceiveFB to update their lists of feedback, and reputations for each user, if the feedback is valid.

Security requirements for PRS

Formally defining the security requirements for a PRS is a complex endeavour, also due to the abundant, differing literature.
So far many models have been put forward, addressing roughly the following requirements.

Security requirements for PRS

Formally defining the security requirements for a PRS is a complex endeavour, also due to the abundant, differing literature.
So far many models have been put forward, addressing roughly the following requirements.

Correctness

This property ensures that if all the algorithms are run correctly, then everything is as "it should".

Security requirements for PRS

Formally defining the security requirements for a PRS is a complex endeavour, also due to the abundant, differing literature.
So far many models have been put forward, addressing roughly the following requirements.

Correctness

This property ensures that if all the algorithms are run correctly, then everything is as ”it should”.

Privacy

This umbrella term may capture untraceability and unlinkability of users, as well as privacy of feedback.

Security requirements for PRS

Formally defining the security requirements for a PRS is a complex endeavour, also due to the abundant, differing literature.
So far many models have been put forward, addressing roughly the following requirements.

Correctness

This property ensures that if all the algorithms are run correctly, then everything is as "it should".

Privacy

This umbrella term may capture untraceability and unlinkability of users, as well as privacy of feedback.

Verifiability

This property ensures only authorised users join the system, and that the reputations are valid.

Attacks on PRS

There are a variety of attacks that can be mounted against PRS [HZN09].

- **Sybil attacks**: users create multiple identities to join the system to give unfair feedback.
- **Unfair ratings attack**: users collude to give unfair ratings.
- **Self-promoting attack**: users manipulate their own reputation by falsely increasing it.
- **Whitewashing attacks**: users leave and rejoin to abuse the system and shed a bad reputation.
- **On-off attack**: users behave honestly to increase their reputation before behaving dishonestly.
- **Reputation lag exploitation**: users exploit the interval before the latest round of ratings takes effect

Anonymous PRS (APRS)

Anonymity has received a great amount of attention in this area.

Anonymous PRS (APRS)

Anonymity has received a great amount of attention in this area.

Anonymous ratings in centralised systems

Solutions exist that

- provide anonymity to all except the reputation server
- provide anonymity using multiple (non-all colluding) servers
- provide anonymity in the presence of a corrupted server
- introduce tokens on anonymous ratings for users to prove their own reputation

The other side of anonymity

Unfortunately, anonymity can provide an opportunity for malicious users to misbehave.

The other side of anonymity

Unfortunately, anonymity can provide an opportunity for malicious users to misbehave.

- Users can "bad mouth" competitors
- Users can collude to inflate their own reputations

The other side of anonymity

Unfortunately, anonymity can provide an opportunity for malicious users to misbehave.

- Users can "bad mouth" competitors
- Users can collude to inflate their own reputations

How do we overcome this?

Typically, the issues raised by anonymity are addressed with the notion of **traceability**.

The other side of anonymity

Unfortunately, anonymity can provide an opportunity for malicious users to misbehave.

- Users can "bad mouth" competitors
- Users can collude to inflate their own reputations

How do we overcome this?

Typically, the issues raised by anonymity are addressed with the notion of **traceability**.

The idea is that if you can trace users when they misbehave, then you can revoke them.

The value of rewards

There is another approach.

The value of rewards

There is another approach.
We can incentivize good behaviour by **rewarding** users.

The value of rewards

There is another approach.
We can incentivize good behaviour by **rewarding** users.
More specifically, by rewarding honest ratings we can achieve *anonymous peer rating systems*, where users are peers and anonymously rate each other.

The value of rewards

There is another approach.

We can incentivize good behaviour by **rewarding** users.

More specifically, by rewarding honest ratings we can achieve *anonymous peer rating systems*, where users are peers and anonymously rate each other.

The rewarding approach works well in contexts where raters are also participating within the system and so have an interest in rating accurately to increase their reputation through rewards.

The value of rewards

There is another approach.
We can incentivize good behaviour by **rewarding** users.
More specifically, by rewarding honest ratings we can achieve *anonymous peer rating systems*, where users are peers and anonymously rate each other.
The rewarding approach works well in contexts where raters are also participating within the system and so have an interest in rating accurately to increase their reputation through rewards.

Examples

- peer-to-peer file sharing
- collaborative knowledge production and sharing

Anonymity and rewards in PRS

In the rest of this talk, we will discuss a solution to APRS which uses rewards [GQNT20].

Anonymity and rewards in PRS

In the rest of this talk, we will discuss a solution to APRS which uses rewards [GQNT20].

- At each round of interaction, users rate each other by providing feedback to the reputation server.
- Accurate ratings are incentivised and weighted by reputation.

Anonymity and rewards in PRS

In the rest of this talk, we will discuss a solution to APRS which uses rewards [GQNT20].

- At each round of interaction, users rate each other by providing feedback to the reputation server.
- Accurate ratings are incentivised and weighted by reputation.

Challenges

- The reputation used to weight feedback could be used to de-anonymise a user.
- Accurate ratings must be incentivised without being de-anonymised.

Anonymity and rewards in PRS

In the rest of this talk, we will discuss a solution to APRS which uses rewards [GQNT20].

- At each round of interaction, users rate each other by providing feedback to the reputation server.
- Accurate ratings are incentivised and weighted by reputation.

Challenges

- The reputation used to weight feedback could be used to de-anonymise a user.
- Accurate ratings must be incentivised without being de-anonymised.

These challenges can be overcome.

Defining APRS: the entities

An APRS comprises the following entities.

- A set of **users**, interacting with each other in rounds.
- The **reputation holder**, a trusted entity which holds the reputations of users.
- The **rewarder**, a newly introduced entity who chooses which ratings to reward, and who cannot see which users have their reputation increase.

We note that the latter two entities can collude or could coincide.

Defining APRS: the interactions

The system we consider operates in *rounds*.

1. At the end of each round users rate each other's performance. In essence, they anonymously send a numerical feedback alongside their reputation to the rewarder.
2. The rewarder collects ratings, discards multiple ratings on the same subject, and rewards accurate feedback by outputting a set of incentives.
3. A user claims to the reputation holder that they were responsible for a number of these incentives.
4. 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.

Defining APRS: the phases I

The definition of an APRS formally consists of 11 algorithms and an interactive protocol.
We shall only discuss its phases here.

Defining APRS: the phases I

The definition of an APRS formally consists of 11 algorithms and an interactive protocol.
We shall only discuss its phases here.

Setup and Key Generation

The reputation holder and rewarder generate their own key pairs.

Defining APRS: the phases I

The definition of an APRS formally consists of 11 algorithms and an interactive protocol.
We shall only discuss its phases here.

Setup and Key Generation

The reputation holder and rewarder generate their own key pairs.

Join

An interactive protocol is run between the users and the reputation holder, providing users with private keys to produce anonymous ratings and collect rewards.

Defining APRS: the phases I

The definition of an APRS formally consists of 11 algorithms and an interactive protocol.
We shall only discuss its phases here.

Setup and Key Generation

The reputation holder and rewarder generate their own key pairs.

Join

An interactive protocol is run between the users and the reputation holder, providing users with private keys to produce anonymous ratings and collect rewards.

Ratings at round l

Users provide ratings which are sent to the rewarder via a secure anonymous channel. The rewarder forms updated reputations, which are securely sent to the reputation holder.

Defining APRS: the phases II

Defining APRS: the phases II

Incentivising accurate feedback

The rewarder rewards accurate feedback by outputting incentives, which are securely collected by the users via an authenticated channel and which influence the users' reputation.

Defining APRS: the phases II

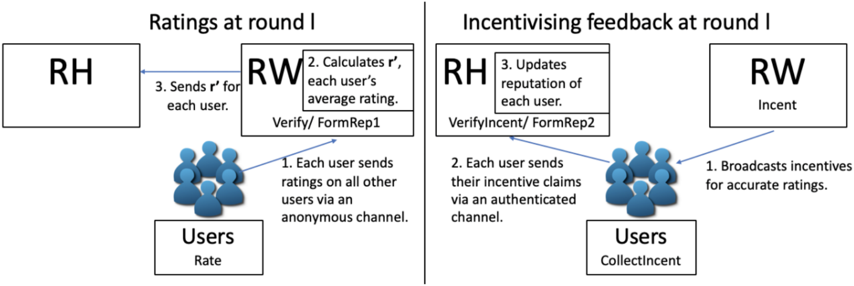
Incentivising accurate feedback

The rewarder rewards accurate feedback by outputting incentives, which are securely collected by the users via an authenticated channel and which influence the users' reputation.

Allocate reputation for next round

A reputation value and a token are given to the user, via an authenticated channel, to prove their reputation in the next round.

Visualizing APRS


Ratings at round l
RH
3. Sends **r'** for each user.
RW
2. Calculates **r'**, each user's average rating.
Verify/ FormRep1
1. Each user sends ratings on all other users via an anonymous channel.
Users
Rate
Incentivising feedback at round l
RH
3. Updates reputation of each user.
VerifyIncent/ FormRep2
RW
Incent
2. Each user sends their incentive claims via an authenticated channel.
1. Broadcasts incentives for accurate ratings.
Users
CollectIncent

Security requirements for APRS

The following correctness and security requirements can be formalised for an APRS.

- **Correctness**
- **Anonymity of Ratings under Full Corruption**, providing the strongest anonymity with both entities corrupted.
- **Anonymity of Ratings under a Corrupt Reputation Holder**, ensuring ratings cannot be de-anonymised or linked by the reputation holder.
- **Traceability**, ensuring multiple ratings cannot be given on the same user per round.
- **Non-Frameability**, ensuring users cannot be impersonated when giving ratings or claiming incentives.
- **Unforgeability of Reputation**, ensuring a user cannot lie about their reputation.
- **Fair Rewards**, ensuring users can only successfully claim for the number of incentives they were awarded.

Building blocks for achieving APRS

To build an APRS, we consider the following primitives.

Building blocks for achieving APRS

To build an APRS, we consider the following primitives.

Public-key Encryption (PKE)

A standard PKE scheme consisting of Setup, KeyGen, Enc and Dec algorithms, and satisfying IND-CCA security.

Building blocks for achieving APRS

To build an APRS, we consider the following primitives.

Public-key Encryption (PKE)

A standard PKE scheme consisting of Setup, KeyGen, Enc and Dec algorithms, and satisfying IND-CCA security.

Linkable Ring Signatures (LRS)

Ring signatures allow users to sign on behalf of a set of users, without revealing their identity within the set. There is no central entity involved, and users generate their own signing and verification keys.

Linkable ring signatures allow for the public linking of signatures by signer.

Building blocks for achieving APRS

To build an APRS, we consider the following primitives.

Public-key Encryption (PKE)

A standard PKE scheme consisting of Setup, KeyGen, Enc and Dec algorithms, and satisfying IND-CCA security.

Linkable Ring Signatures (LRS)

Ring signatures allow users to sign on behalf of a set of users, without revealing their identity within the set. There is no central entity involved, and users generate their own signing and verification keys.
Linkable ring signatures allow for the public linking of signatures by signer.

Direct Anonymous Attestation (DAA)

DAA, similarly to RS, allows users to sign on behalf of a group, whilst remaining anonymous within the group. However, there is a central authority involved.

A construction for APRS [GQNT20]

In [GQNT20] a construction for APRS based on PKE, LRS and DAA is provided.

A construction for APRS [GQNT20]

In [GQNT20] a construction for APRS based on PKE, LRS and DAA is provided.

- The construction makes use of DAA to **sign feedback**.
- The DAA scheme is modified so that when giving feedback a user can prove they have a particular reputation for that round, so that **feedback can be weighted**.
- LRS are used 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.

Security of [GQNT20]

Security of [GQNT20]

- Anonymity of Ratings under Full-Corruption stems from the anonymity properties of LRS and DAA.

Security of [GQNT20]

- Anonymity of Ratings under Full-Corruption stems from the anonymity properties of LRS and DAA.
- Traceability: The User Controlled Linkability of DAA ensures multiple feedback cannot be given per subject.

Security of [GQNT20]

- Anonymity of Ratings under Full-Corruption stems from the anonymity properties of LRS and DAA.
- Traceability: The User Controlled Linkability of DAA ensures multiple feedback cannot be given per subject.
- Non-Frameability and Unforgeability of Reputation derives from analogous properties in LRS and DAA.

Security of [GQNT20]

- Anonymity of Ratings under Full-Corruption stems from the anonymity properties of LRS and DAA.
- Traceability: The User Controlled Linkability of DAA ensures multiple feedback cannot be given per subject.
- Non-Frameability and Unforgeability of Reputation derives from analogous properties in LRS and DAA.
- Fair Rewards: Due to the linkability of LRS, multiple claims for the same rating can be detected.

Security of [GQNT20]

- Anonymity of Ratings under Full-Corruption stems from the anonymity properties of LRS and DAA.
- Traceability: The User Controlled Linkability of DAA ensures multiple feedback cannot be given per subject.
- Non-Frameability and Unforgeability of Reputation derives from analogous properties in LRS and DAA.
- Fair Rewards: Due to the linkability of LRS, multiple claims for the same rating can be detected.
- The fact that a central entity, i.e., the reputation holder, must authorise the creation of a new DAA key pair, prevents sybil attacks.

Security of [GQNT20]

- Anonymity of Ratings under Full-Corruption stems from the anonymity properties of LRS and DAA.
- Traceability: The User Controlled Linkability of DAA ensures multiple feedback cannot be given per subject.
- Non-Frameability and Unforgeability of Reputation derives from analogous properties in LRS and DAA.
- Fair Rewards: Due to the linkability of LRS, multiple claims for the same rating can be detected.
- The fact that a central entity, i.e., the reputation holder, must authorise the creation of a new DAA key pair, prevents sybil attacks.

Detailed discussions regarding other types of attacks can be found in [GQNT20].

Instantiation and Efficiency

Instantiation and Efficiency

Instantiation

The construction is a generic design and can be instantiated with existing schemes under standard assumptions (e.g., DDH and Random Oracle).

Instantiation and Efficiency

Instantiation

The construction is a generic design and can be instantiated with existing schemes under standard assumptions (e.g., DDH and Random Oracle).

Efficiency

- An incentive claim would have size log(ℓ)poly(τ), 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.

Conclusion and future work

Conclusion and future work

- The current state of the art achieves APRS 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*.

Conclusion and future work

- The current state of the art achieves APRS 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*.
- The construction provided is generic, so different primitives could be used to build systems that are more efficient or rely on different assumptions.

Conclusion and future work

- The current state of the art achieves APRS 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*.
- The construction provided is generic, so different primitives could be used to build systems that are more efficient or rely on different assumptions.
- It would be interesting to study APRS through a game theoretic approach, where strategies and incentives play a fundamental role.

De Componendis Cifris

https://www.decifris.it

Main References in this Talk

- [HZN09] *A survey of attack and defense techniques for reputation systems*, Kevin Hoffman, David Zage and Cristina Nita-Rotaru, *ACM Computing Surveys 2009*.
- [GQ19] *A new approach to modelling centralised reputation systems*, Lydia Garms and Elizabeth A. Quaglia, *AFRICACRYPT 2019*.
- [GQNT20] *Anonymity and Rewards in Peer Rating Systems*, Lydia Garms, Siaw-Lynn Ng, Elizabeth A. Quaglia and Giulia Traverso, *SCN 2020*.