A Reputation System supporting Unlinkable, yet Authorized Expert Ratings

Andreas Kokoschka
Computer Science Dpt.
University of Paderborn
33098 Paderborn, Germany
andiko@mail.upb.de

Ronald Petrlic*
CISPA
Saarland University
66111 Saarbrücken, Germany
ronald.petrlic@uni-saarland.de

Christoph Sorge
CISPA
Saarland University
66111 Saarbrücken, Germany
christoph.sorge@uni-saarland.de

ABSTRACT

Reputation systems used in practice typically either provide *robustness* or *anonymity*. A lot of research has been going on to come up with schemes that provide both properties, however most of them being too impractical. We come up with an approach for a reputation system that provides anonymity for users, meaning that ratings cannot be linked to raters, but at the same time a rater's identity can be disclosed in case a service is rated twice by a user—having the permission to perform only a single rating. This is achieved by making use of a group signature variant, whose properties are described in detail as well. Moreover, we aim to make our system "lively" by introducing the concept of *expert raters*, which shall constitute an incentive for users to actively participate in the reputation system by providing ratings. We believe that this functionality is an important one towards *practicability*.

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and Protection—*Authentication, Unauthorized access*; K.4.1 [**Computers and Society**]: Public Policy Issues—*Privacy*

Keywords

Reputation Management, Privacy Protection, Anonymity

1. INTRODUCTION

A major difference between traditional business and electronic marketplaces is the way trust in transaction partners is established. Electronic business is often conducted with previously unknown entities. Assessing their trustworthiness based on physical appearance is impossible, and word of mouth is not usually available. Reputation systems fill that gap: transaction partners can rate each other, and assess the respective trustworthiness based on (aggregates of) others' ratings. Unfortunately, reputation systems may constitute a privacy risk, as the raters are usually identified—at least under a pseudonym. Detailed profiles can be created by anyone with access to the (non-aggregated) ratings.

As pseudonymous profiles can often be mapped to persons, we aim at designing an anonymous reputation system: Users can provide their ratings without having to reveal any personally-identifying information, and it shall not be possible for any party to link ratings to transactions or ratings to each other: *unlinkability* is achieved. Robustness is achieved if it is not possible for any user to rate a transaction twice. In our scenario, robustness means that if a user succeeds in rating a transaction twice, he will get caught.

1.1 Contribution

Our contribution is to design an *anonymous* and *robust* reputation system. These two requirements are contradictory at first glance, but we show that by making use of *group signatures*, both properties can be achieved at the same time. Moreover, our reputation system provides support for raters with different expertise—allowing users, when looking at the reputation values, to place higher confidence in ratings by experts, for example. Our system requires a group signature scheme with properties that are not fulfilled by current schemes, but we provide evidence indicating such a system can be constructed.

1.2 Paper Outline

The remainder of this paper is structured as follows: In Section 2, we discuss preliminaries, like the cryptographic building blocks to be used. Section 3 describes the abstract model of our reputation system, while Section 4 deals with the concrete concept. We evaluate our approach in Section 5 before comparing it with related work in Section 6. The paper is concluded in Section 7.

2. PRELIMINARIES

2.1 Reputation Systems

Reputation systems provide information used to judge the trustworthiness of (potential) transaction partners. If a system simply exposed all ratings and the identities of the respective raters, any user could decide for himself which ratings to consider trustworthy, and how to aggregate them. However, to achieve anonymity, some information must be removed or aggregated by the system; it is an important

*Corresponding Author

SAC'15 April 13-17, 2015, Salamanca, Spain.
 ACM 978-1-4503-3196-8/15/04...$15.00
http://dx.doi.org/10.1145/2695664.2695892

design decision which information can still be exposed. In particular, we consider the rater's experience (i.e. an indication about the number of previously rated transactions) as a relevant criterion, which we make available to the users.

A number of issues have to be considered in reputation systems that work either without identities, or with cheap identities. *Sybil attacks* are a common problem, meaning that one party appears under multiple identities to skew aggregate ratings (also referred to as ballot stuffing, a term originating from voting systems). This can be used both for self-promoting and for slandering attacks. *Whitewashing* is a related problem: A rated entity can create a new identity so old, negative ratings can no longer be attributed to it. For an extensive discussion of attacks on and defenses of reputation systems, see Hoffman [11].

2.2 Cryptographic Building Blocks

In this section, we describe some of the cryptographic building blocks used in our reputation system.

2.2.1 Partially Blind Signatures

Brands [4] introduces the concept of "restrictive blind signatures", in which not all the information is blinded, as in other blind signature schemes, like e.g. the one by Chaum et al. [5]. This property allows any verifier to derive some additional information for verification.

Abe et al. [1] construct a method to "date" blind signatures, which allows including arbitrary public data in a blind signature. Signature schemes with that property are commonly known as "partially blind signatures".

The actual scheme that is used in this paper is due to Chien et al. [7]. The scheme is based on RSA and requires very low computational effort. To generate a signature, four algorithms are executed by signer and signature requester. The processes are as follows:

Initialization:

- The signer sets up public key (e, n) and private key (d, n) according to the RSA scheme
- The signer publishes his public key and defines a secure one-way hash function h to be used

Request:

- The requester selects the common information a to be included in the signature for message m
- The requester chooses the blinding factors r and $u \in \mathbb{Z}_n^*$ uniformly at random
- The requester sends (a, α) to the signer, where $\alpha = r^e h(m)(u^2 + 1) \mod n$
- The signer verifies a and responds with a random integer $x \leq n$
- Let r' be an integer value chosen uniformly at random and $b = r \cdot r'$
- The requester computes $\beta = b^e(u - x) \mod n$ and sends β to the signer

Signing:

- Upon receiving β, the signer computes $t = h(a)^d(\alpha(x^2 + 1)\beta^{-2})^{2d} \mod n$
- The blind signature (β^{-1}, t) is sent back to the requester

Extraction:

- The requester computes a valid signature with $s = t \cdot r^2 \cdot r'^4 \mod n$ and $c = (ux + 1) \cdot \beta^{-1} \cdot b^e = (ux + 1)(u - x)^{-1} \mod m$

The resulting signature can finally be verified by checking whether the equation $s^e = h(a)h(m)^2(c^2+1)^2 \mod n$ holds.

2.2.2 Group Signatures

Group signatures were introduced by Chaum et al. [6]. Those schemes allow an arbitrary member of a group to anonymously sign messages on behalf of the group. In a first step, the group members need to enroll with a dedicated trusted group manager. The group manager decides whether the new member is allowed to join the group and issues a private group signing key for the new member. It is also the group manager who is able to revoke the anonymity of a single group member by exposing him as the one responsible for a particular signature. Group signatures do not only provide unlinkability of signatures to their authors, but also guarantee unlinkability of two different signatures.

Boneh et al. [3] present a group signature scheme based on bilinear maps, which allows for very short signatures and low computational effort for signature verification. Boneh et al. also discuss revocation mechanisms and conclude that ideally, only a broadcast to all signature verifiers is needed. Their proposed signature scheme also provides that functionality, which is called *verifier-local-revocation (VLR)*. However, the revocation mechanism leads to a computational cost, increasing linear with the number of revoked group members.

2.2.3 Ring Signatures and One-more Unforgeability

Ring signatures provide similar properties to group signatures, without the requirement for a group manager. Thus, there is no functionality for anonymity revocation. A ring signature can be computed by any member of a group of users if the public keys of all the group members are accessible beforehand. It is not possible to link any group member to a concrete signature, thus, the determination of a signer of a message is not possible. There exist a number of ring signature schemes in the literature.

The ring signature scheme by Fujisaki et al. [10] provides an interesting property, which is also required in the group signature scheme used in the paper at hand. They propose the concept of *traceable* ring signatures. For that purpose, "tags" are introduced for every signed message. A tag $T = (t, pKN)$ consists of a tag name t and the total set of public keys of all ring members pKN. Every ring member can sign messages with his respective private key, including a tag T' with an arbitrary tag name. Messages can be verified with respect to the included tag, while the identity of the signer remains private as usual for any kind of ring signature. The distinct property of the traceable signature scheme are two distinguishable attributes provided:

- Every two signatures with the same tag but different messages can be traced by any verifier with only knowl-

edge of both messages, both signatures and the common tag (public traceability)

- Every two signatures generated by a single signer and including the same tag are linked (tag-linkability)

The second property ensures the so-called "one-more unforgeability", as no tag can be used more than once by any ring member for different messages without making the multiple usage visible to any verifier.[1]

2.2.4 One-more Unforgeability for Group Signatures

As of today, there exist no group signature schemes with attributes comparable to those of ring signature schemes as discussed above.

However, DAMGÅRD ET AL. [8] present a group signature variant which provides *one-more unforgeability*. The scheme relies on *zero-knowledge proofs* [9] performed for every signature that a group member generates. Like the traceable ring signature, it can expose the identity of members who sign two different messages with a respective value α. This value can be considered equivalent to the tags discussed before. However, the signature generation requires communication with the group manager. Setting the parameters so that the scheme offers "reasonable" security, the total amount of data exchanged between the group manager and a group member for signature generation sums up to about 130 Kilobytes.

3. SYSTEM MODEL

Our proposed reputation system comprises of the following entities:

- Reputation provider which stores the users' ratings and provides the service providers' reputation values to users
- Service providers which are rated by users
- Users who rate service providers and retrieve service providers' reputation values
- Group manager which is responsible for registering the users who use the group signature scheme
- Prestige manager which is responsible for managing the users' "prestiges", i.e. the users' expertise levels

Users rate service providers—not services/transactions—after having performed a transaction with them.

3.1 Requirements

The requirements to our system can be classified into the following categories.

3.1.1 Liveliness

Liveliness is a very important requirement in reputation systems—neglect often leading to reputation systems that pose no practical advantage as ratings are spare and, thus, reputation values simply not being available or not significant. This is the reason why we strive for a mechanism that rewards users who provide a rating for a completed transaction. Liveliness is achieved if the following properties are met:

- Rating Incentive: The system shall offer a rating incentive for users in order to maintain a high level of ratings.
- Quick Rating: Prompt acquisition of recent ratings in order to have up-to-date reputation values and to avoid negative ratings by users after the time for complaints has passed.
- Reflection of up-to-date State: The service provider's reputation value must not show a high inertia if the average ratings drop quickly.

The incentive for providing ratings in our scenario is supposed to be achieved by a mechanism that shall allow users to excel, i.e. to gain "prestige" by submitting ratings and become an "expert rater" at some point of time. Expert raters' ratings, on the other hand, may be valued more by other users as they can assume that those expert raters—who have performed a lot of transactions already—know better how to assess transactions (service providers) and, thus, rating them more accurately than others could do. The requirement for expert ratings can also be found in the literature [12].

Moreover, many reputation systems—especially those that provide privacy by not revealing individual ratings but only an aggregated reputation value—face the problem that they do not adapt well to new situations. For example, if a service provider delivered good quality over a long period of time, his reputation value was, let us assume, good. However, at a certain point of time, the service provider decides to just deliver bad quality from now on, resulting in, let us assume, bad ratings. However, those (current) bad ratings might not have a strong impact in some reputation systems as only the aggregation over all—including the many good—ratings is made available as reputation value. We thus strive for a reputation system that does not have this problem.

3.1.2 Security

If ratings for service providers were to be submitted by arbitrary users—even those not having dealt with the service providers—, those ratings, and the reputation values based upon them, would be meaningless. Moreover, legitimate ratings and reputation values should not be alterable by any party. Security is achieved if the following sub-requirements are met:

- Authorization: Only after having performed a transaction with a service provider, a user shall be able to rate that service provider
- Integrity: No ratings may be altered during transmission or while being stored at the reputation provider

3.1.3 Privacy

The main motivation for users' privacy to be adhered to in a reputation system is that users are expected to rate more honestly—and, rate at all. Users might not rate service providers that provide products they do not want to be connected to by other users, and, thus, would not use a reputation system that relies on identification during ratings. Privacy is achieved if the following protection goals are met:

- Anonymous Rating: Users shall be able to transmit their ratings to the reputation provider without revealing any personally-identifiable information. For that

[1]"The total number of signatures with respect to the same tag cannot exceed the total number of ring members in the tag, if every any two signatures are not linked." [10]

purpose, the following unlinkability guarantees need to be met:

- Unlinkability of ratings to transactions
- Unlinkability of raters to transactions
- Unlinkability of any two or more ratings

3.2 Assumptions

A reliable reputation system is based on the authenticity of its users' identities. In consequence, such a system requires sufficient cost for users' identities. We thus assume, that an entity already exists that can verify the identity of any user. Furthermore, we assume that a public key infrastructure (PKI) is accessible and all users hold verifiable credentials or can obtain them at any time, providing their verified identity. Acquisition of those credentials is assumed to impose sufficiently high cost for all users in order to prevent *Sybil attacks*.

In order for users to stay fully anonymous, *communication anonymity* also needs to be given. Therefore, we assume users to connect to the reputation system through mechanisms such as Tor.

Communication channel security (i.e. secrecy, integrity, and authenticity) is assumed to be given by using standard mechanisms such as TLS.

Moreover, we assume a group signature scheme with one-more unforgeability that provides the following properties:

- After enrollment at the group manager, the generation of a group signature can be done individually by every group member without depending on any other entity.
- The group manager can trace any signature and reveal the identity of the signer.
- Every group signature contains a tag T, with T being an arbitrary value.
- Every two signatures generated with a single group member's secret key and including the same tag are linked (one-more unforgeability).
- The signature scheme is fully dynamic, allowing users to join the group at any time after initializing and providing a method for member revocation.
- The signature scheme offers verifier-local-revocation [3] with a logarithmic or better correlation between the number of revoked group members and computational cost for verification.

3.3 Definition of Security and Privacy

We assume that there is no single trusted third party (TTP) that keeps track of reputation information, i.e. particularly the reputation provider is not a TTP. The reputation provider is modeled as a *malicious adversary* that might deviate from the protocol specification.

The reputation provider must not learn the real identity of the user submitting a rating—unless the user rates the same service provider twice.

Furthermore, the reputation provider shall not be able to manipulate any ratings, i.e. not publish them when providing reputation values.

The service providers shall not learn which users provided ratings for their services. If there is only a small number of users who performed a transaction (and, thus, a small number of ratings), the service provider shall not learn those ratings as it would allow drawing conclusions about which user rendered in a certain rating—especially if the user reveals his real identity for the transaction, e.g. for orderings of physical goods.

4. CONCEPT

The use cases of the system are shown in Fig. 4. The individual phases of the overall system are described next in more detail.

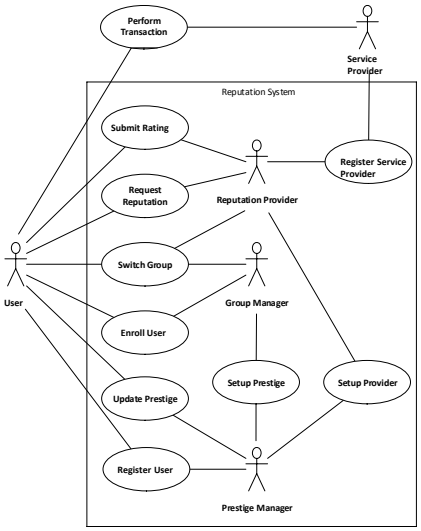

Figure 1: Use cases of the proposed reputation system

4.1 Setup Prestige

The *prestige manager* sets the users' initial amount of prestige $prestige_{init}$, as well as a value $prestige_{add}$ by which the user's prestige is incremented for every rating he submits. Then the prestige levels $Prestige_1, \ldots, Prestige_k$ are defined and a trusted *group manager* is selected to initialize the groups $g_1, \ldots, g_k$ with group IDs denoted by $G_1, \ldots, G_k$—corresponding to the prestige levels. After the group generation is completed, the prestige manager generates a private key pm_{priv} and the corresponding public key pm_{pub}, which is then forwarded to the group manager. This key is needed to verify the partially blind signed token, which every user needs to provide for enrollment at the group manager.

To allow for a promotion of the users during the operational period, a threshold vector $V_{threshold} = (v_1, \ldots, v_k)$ is generated—k equaling the total number of groups generated before. A user's amount of prestige $prestige_u$ needs to be greater or equal to v_i in order to be promoted to group G_i.

4.2 Setup Reputation Provider

In the first step, the reputation provider needs to select the desired prestige manager. As rating aggregation can be dependent on the prestige levels specified by the prestige manager, aggregation functions should be defined at this point. Moreover, the structure of ratings to be accepted needs to be published—in order to be accessible for

everyone—beforehand as well. For arbitrary ratings, a general structure in form of a vector R_{tr} can be assumed. The number of ratings C that are collected before publishing them—as reputation value—, is also set.

Finally, the reputation provider generates a private key rp_{priv} and the corresponding public key rp_{pub}, which is forwarded to the prestige manager with a request for registration. If the prestige manager accepts the request, the system is ready to accept ratings.

4.3 Register Service Provider

The service provider generates a private key sp_{priv} and the corresponding public key sp_{pub}, which is forwarded to the reputation provider. If the service provider agrees to the processing and sharing of its reputation value, the reputation provider issues an ID ID_{sp} and a token-time[2] TT_{sp} to the service provider.

4.4 Register User

While users are not required to register at the reputation provider, a registration at the prestige manager is required. This registration can be performed without exposing users' identities. The goal of registering at the prestige manager is to gain credentials that allow the reputation provider to identify the respective user's level of prestige and prevent providing ratings several times.

In the first step, the user generates his private key u_{priv} and the corresponding public key u_{pub}, which he forwards to the prestige manager. The prestige manager generates a new account a_u with initial prestige $prestige_{init}$ and stores the received public key u_{pub} associated with that account. Then the prestige manager forwards the user's new account number A_u and the prestige manager's public key pm_{pub}.

The user then acquires a partially blind signature for a random token T_{ru}. This process, which provides no knowledge of T_{ru} to the signer but reveals the ID G_1 of the initial group as part of the signature, works as follows. The user computes the hash value $h(T_{ru})$ and blinds the value with the received public key pm_{pub} to get $h(T_{ru})^*$. This result is sent to the prestige manager who returns a blind signature $\sigma_{pm}(h(T_{ru})^*, G_1)$ for $h(T_{ru})^*$ with the group ID as public information. The user then unblinds the result to obtain a valid signature including the ID of the initial group. $(T_{ru}, G_1)'$ denotes the signed value of T_{ru}.

4.5 Enroll User

Every user needs to become a member of one of the groups generated during the setup of the reputation provider before being able to submit ratings. To obtain membership of a group, the user needs to submit his credentials to the group manager. If the group manager accepts the credentials, he first checks whether the user has already enrolled before. If that is not the case, the group manager asks the user for the token $(T_{ru}, G_1)'$, which has been blindly signed by the prestige manager during the registration process. As described previously, $(T_{ru}, G_1)'$ includes the ID of the initial group g_1. This informs the group manager about the group the respective user is supposed to enroll in. The user provides that token and the group manager then checks the received signature for validity. If it is valid, the list of all previously used tokens is searched for the user's token. In case it has not been used before, it is added to the list. The user's credentials are then stored and the user is enrolled.

At the end of this process, the group manager issues a valid group secret key GSK_u to the user. The user stores all information, as they will be used later on when he wants to enroll in another group.

4.6 Request Reputation

Any user can request service providers' reputation values, by asking the reputation provider for the reputation value of the service provider with ID ID_{sp}. The user can specify the exact reputation information desired. It can either be the total amount of ratings $R_{total}(ID_{sp}, (t_{begin}, t_{end}))$ over a period of time (t_{begin}, t_{end}) or any form of aggregated data $R_{agg}(s, (t_{begin}, t_{end}))$ which the reputation provider is willing to provide for some respective period.

Given the possibility to access all "raw" ratings, any user is thus able to verify the contained signatures and to decide whether a concrete reputation value is trustworthy.

4.7 Perform Transaction

Any transactions between users and service provider are supported by our reputation system. Before processing any payment for the transaction, though, the service provider needs to supply the user with its public key sp_{pub} and the user asks for a (blindly) signed token, which works as follows. The user generates a random token T_r. Then he computes its hash value $h(T_r)$ and blinds the result using the seller's public key sp_{pub} to receive a partially blind signature for the resulting value $h(T_r)^*$ once again. The common information is the service provider's public token-time TT_{sp}. The user forwards $h(T_r)^*$ to the service provider who responds with the desired blind signature $\sigma_{sp}(h(T_r)^*, TT_{sp})$. The user unblinds the received signature and eventually holds the signed value $(T_r, TT_{sp})'$ for his random token generated before. If the signature passes verification, the user processes the required payment and the transaction is carried out.

4.8 Submit Rating

As covered above, users are granted with a signed random token $(T_r, TT_{sp})'$ of their own choice during transactions, which entitles them to submit ratings to the reputation provider and, furthermore, increase their prestige after submission. Moreover, as we have seen, the user is in possession of a valid group secret key GSK_u for the group G_u, where G_u denotes the group the user is enrolled in.

To submit a rating to the reputation provider, the user sends his token $(T_r, TT_{sp})'$ first. The signature of the token is verified and the token is checked for double-spending. If the token is not found on the list of all previously submitted tokens, the token is added to the list. Moreover, the reputation provider checks whether the service provider's token-time is still valid. Let TT'_{sp} be the service provider's current token-time. If $TT'_{sp} \geq TT_{sp} + 1$ holds, the token's lifetime has expired and, thus, the token is rejected. Otherwise, the reputation provider acknowledges the token's validity and requests for the user's rating.

The user generates a rating vector R_{tr}. He signs R_{tr} using his group secret key GSK_u and the required tag ID_{sp}, giving the signed value $(R_{tr}, ID_{sp})'$. Note that ID_{sp} is the respective service provider's public ID. $(R_{tr}, ID_{sp})'$ is then transmitted together with the user's group ID G_u.

The reputation provider checks the validity of the group

[2]Token-time means that the token includes a time-stamp.

signature for the claimed group g_u. If the signature is valid, the rating is accepted. Given the group signature scheme's property of one-more unforgeability, any two vectors of different ratings which are signed with a single user's key are inevitably linked. The set of received ratings is therefore searched for any linked values. If some linked ratings are found, the current process is aborted. Otherwise, the reputation provider adds $(R_{tr}, ID_{sp})'$, G_u, $(T_r, TT_{sp})'$ as one element to the set W_{sp}, denoting the ratings accumulated for service provider ID_{sp} and waiting for disclosure. If the amount of tokens with token-time TT'_{sp} exceeds C, TT'_{sp} is increased by one. All ratings with token-time $TT_{sp} \neq TT'_{sp}$ are released in random order, if W_{sp} contains more than C elements. This data then reveals the particular signed ratings, i.e. the reputation value, the associated users' levels of prestige and the users' authorization for ratings.

To reward the user for his rating submission, the reputation provider transmits his public key rp_{pub} to the user and the user once again generates a token, which will be (blindly) signed by the reputation provider—this token can then be presented to the prestige manager to get an increased prestige. This process works as follows. The user acquires a partially blind signature with his group ID G_u, denoting his current prestige level. The user generates a random token T_p, computes the hash value $h(T_p)$ and blinds the result using the reputation provider's public key rp_{pub}. The resulting value $h(T_p)^*$ is transmitted to the reputation provider who responds with the desired blind signature $\sigma_{rp}(h(T_p^*, G_u)$. The user unblinds the received signature to get the signed value $(T_p, G_u)'$ of his token T_p, with his prestige level included in the signature. The token is verified by the user.

4.9 Update Prestige

The partially blind token $(T_p, G_u)'$, received during the rating submission, entitles the user to increase his prestige at the prestige manager. Therefore, the user signs in to his account a_u at the prestige manager and submits the token. The prestige manager verifies the validity of the signature and the group index G_u. If the check succeeds, the user's current prestige value $prestige_u$ is increased by $prestige_{add}$ and the new value is returned.

The prestige value offers promotion to a higher prestige level. Thus, after incrementation, it is checked whether the value reaches a certain threshold. If that is the case, the user is promoted to the next level. Let $prestige_u$ be the user's current prestige level. The respective threshold for promotion to the next level is then given by v_{u+1}, defined in the threshold vector $V_{threshold}$. Thus, if $v_{u+1} \leq prestige_u$, the user is promoted. As the prestige level is associated to the user's group membership, switching the group is required as well.

To switch the group, a partially blind signature is generated. The common information contained in the signature is a vector $\alpha = (G_u, G_{new})$. It contains the IDs of the user's current and future group.

A random token $T_{promotion}$ is generated by the user who then computes its hash value $h(T_{promotion})$ and blinds the result using the prestige manager's public key pm_{pub}. The blinded value $h(T_{promotion})^*$ is transmitted to the prestige manager. The prestige manager computes the blind signature $\sigma_{pm}(h(T_{promotion})^*, \alpha)$ and forwards it to the user. The user unblinds the received signature to get the signed value $(T_{promotion}, \alpha)'$ of his token $T_{promotion}$.

4.10 Switch Group

To switch to a group representing a higher prestige level, a user has to contact the group manager. The user transmits his credentials provided for enrollment before and the group manager verifies them. If the verification succeeds, the group manager asks for a token, authorizing the switch to another group. The user sends his signed token $(T_{promotion}, \alpha)'$, which is verified by the group manager. As $\alpha = (G_u, G_{new})$ defines the group member's claimed current group, the group manager can verify this information against his own data. If the token matches and cannot be found in the list of previously submitted tokens, the group manager revokes the user's current group secret key GSK_u and informs the reputation provider to update his revocation list. The user is then enrolled in the group corresponding to his new prestige level and $(T_{promotion}, \alpha)'$ is added to the list of previously used tokens. Transmission of the user's new group secret key GSK'_u finishes the procedure.

5. EVALUATION AND DISCUSSION

In this section we check whether the requirements as stated in Sect. 3.1 are met. We begin with the general requirements to the reputation system in terms of functionality and then show that the proposed concept is secure and privacy-preserving.

5.1 Liveliness

In comparison to related work on privacy-preserving reputation systems, our proposed concept also puts a focus on the liveliness of the reputation system, i.e. that the reputation system is actively used by users.

5.1.1 Rating Incentive

In reputation systems used nowadays, users do not have any direct incentive to provide ratings: it costs time to provide a rating but it helps only other users. However, a reputation system with a high number of ratings is more reliable, i.e. the reputation value is more accurate. This is why we have introduced what we call "prestige" in our concept. The prestige levels of users increases with all submitted ratings. Users thus get the chance to ascent and become "expert raters", which means that they are more appreciated by other users. At the same time, users are not able to cheat, as we will cover in the security analysis: ratings can only be provided if transactions actually took place. Thus, we believe that our approach of introducing prestige provides a good way of rating incentive and will make our reputation system "livelier".

5.1.2 Quick Rating

As we have shown in our concept description, ratings can only be transmitted for a certain period of time after a transaction has taken place. This approach protects services providers from being threatened by users long after the transaction. If users were able to provide a rating even after a long time, the rating could serve as some sort of "insurance" for the user: If the user is not happy with the product any longer after one year, for example, he could simply demand his money for the product back and threaten to provide a very bad feedback if the service provider does not comply with the demand. Especially in a reputation system where individual ratings have a high impact on the reputa-

tion value, a service provider would not take that risk of not complying with the users' demands.

5.1.3 Reflection of up-to-date State

Inertia is a property that most anonymous reputation systems struggle with. The reason for that lies in the trade-off between anonymity by hiding in an anonymity set and quick provision of reputation values. With the size of the anonymity set in the proposed concept set to C, there is a lower bound imposed for the reputation's inertia. Except for this, there is no additional delay as ratings can also be accessed individually, i.e. as "raw data", which will quickly show rapid reputation changes—e.g., if the service provider starts to deliver only bad quality from a certain point of time on.

5.2 Security Analysis

Our proposed reputation system conforms to the security requirements that hold for most reputation systems in use today as well.

5.2.1 Authorization

As we have seen, users get supplied with an authorization token that entitles for the submission of a rating during the transaction from the service provider. Only users in possession of such tokens are able to rate a service provider—only once. The communication channels provide confidentiality. Thus, the reputation provider is the only entity, besides the user, which gets access to the token. In theory, the reputation provider would be the only entity that could "steal" that token from the user and use it to submit a rating with the user's prestige. However, the user would immediately identify such fraudulent behavior as all ratings are made public. Furthermore, the user can prove ownership by revealing the blinding factor used for acquisition of the signature.

5.2.2 Integrity

Integrity of ratings and, thus, reputation values, is provided on two levels. First, all ratings are publicly accessible and therefore verifiable (by the raters themselves). Secondly, integrity is based on the unforgeability property of the underlying group signature scheme: Generation of a valid signature for altered ratings is computationally infeasible. Thus, no ratings can be altered without invalidating the group signature.

5.3 Privacy Analysis

Privacy protection is a fundamental requirement for reputation systems to provide users the possibility to rate transactions/service providers they would otherwise not do, as we have argued.

In contrast to reputation systems used in practice, and also related work in this field, we argue that the sole use of pseudonyms is not enough for users to stay anonymous. It has been shown that profiles under a pseudonym can be de-anonymized, for example. Thus, we required fully anonymous ratings that are unlinkable to each other in Sect. 3.1.

5.3.1 Anonymous Rating

Recapitulating the rating transmission presented in Sect. 4, we can see that a distinct set of data is transferred to the reputation provider: $((R_{tr}, ID_{sp})', G_u, (T_r, TT_{sp})')$. As this set is eventually, i.e. after publishing as reputation value, visible to everyone, an adversary with cumulative knowledge must be considered to protect against.

It is to be shown now, that

- unlinkability of ratings to transactions
- unlinkability of raters to transactions
- unlinkability of any two or more ratings

is given. Therefore, let us look at the data an attacker knows in detail. The rating itself, R_{tr}, does not allow the attacker a linkage to a transaction or a rater. In theory, it might be possible for the service provider to link a bad feedback to a bad quality service provided—given, the bad service was provided only once. However, no "meaningful" information can be derived from that in practice. Every reputation system is prone to that very theoretic attack. The service provider's ID, ID_{sp}, does not provide any information for linkage either. With G_u derivable from the group signature, there is only the group signature itself to come into consideration, though any information gained here would invalidate the respective cryptographic assumptions. The prestige manager might try to arbitrarily promote a target user to a unique group, but in practice, this will catch the group manager's attention. The blindly signed token, $(T_r, TT_{sp})'$, does not reveal anything apart from the token-time TT_{sp}. This is assumed to hold due to the proven security of the blind signature scheme. The only entity with knowledge of the transaction is the service provider. As the blindly signed token reveals no data to the service provider, the token-time is the only hint on the underlying transaction. Since the reputation provider is not disclosing any rating before there are at least C values for a single token-time, this results in an anonymity set with C elements. Moreover, as group signatures are unlinkable to each other—as long as they do not contain the same tag—ratings are unlinkable to each other as well, and, thus, no profile under a pseudonym can be built. To sum it up, if users use an anonymization network such as Tor, as assumed in this paper, users can stay *fully anonymous* when using our proposed reputation system.

6. RELATED WORK

Androulaki et al. [2] propose a reputation system concept offering anonymous ratings. However, there is no direct mechanism for rating authorization. In addition, their design does not provide full robustness against collaborating attackers, assuming that they never share private keys. Schiffner et al. [15] added rating authorization in their approach, but they do not consider ballot-stuffing, as their concept can only rely on a general fee for payment processing to repel fake transactions.

Kerschbaum [13] presents a reputation system that provides anonymity in the virtual organization formation scenario. Ratings are coupled to transactions and transaction partners can rate each other. In his approach, there are two (centralized) mutually mistrusting reputation providers RP_1 and RP_2 with different tasks. To rate a transaction partner, a user encrypts his rating and sends it to RP_1. RP_1 collects a number of individual ratings before posting them in a block to a bulletin board. RP_2 takes the ratings from the bulletin board, decrypts and aggregates them and provides them for retrieval. The scheme is based on the homomorphic Paillier cryptosystem. The usage of two reputation providers

guarantees the unlinkability of rating users and ratings. By using the bulletin board, unlinkability between ratings and transactions is achieved. PETRLIC ET AL. [14] show that a reputation system based on homomorphic encryption is also possible by making use of only a single reputation provider—which does not need to be fully trusted. However, a drawback of approaches based on homomorphic aggregation is that arbitrary ratings are not supported. Moreover, the systems do not provide robustness against ballot-stuffing either.

7. CONCLUSION AND OUTLOOK

In this paper we have made several contributions. We came up with a proposal for a reputation system that combines *security*, *privacy*, and *liveliness*. In our reputation system, only users who are authorized to rate transactions can do that—at the same time, users stay anonymous during those ratings, i.e. no party can link a rating to an individual user. By making use of a group signature scheme with the one-more unforgeability property, it is possible to detect fraudulent users, i.e. those users who provide several ratings for a transaction they are entitled to rate only once. This ensures *robustness* of our reputation system as users can not rate arbitrary transactions they have not performed—as it is the case for other reputation systems. Moreover, our proposed reputation system also guarantees *liveliness*. That said, users have an incentive to actively participate in our reputation system. For every rating users provide, their "prestige", i.e. their level of expertise, increases. Active raters can thus become "expert raters", which means that they are more appreciated by other users at the end. Other users, in turn, can put more value to ratings provided by expert raters if they want. This is made possible because the reputation value that is retrieved is made up of individual ratings that can be retrieved—which is different to privacy-preserving reputation system approaches in the literature. In other approaches, only an aggregation of ratings can be retrieved.

For future work, it would be interesting to see more research on group signature schemes that provide one-more unforgeability.

8. REFERENCES

[1] Masayuki Abe and Eiichiro Fujisaki. How to date blind signatures. In Kwangjo Kim and Tsutomu Matsumoto, editors, *Advances in Cryptology (ASIACRYPT '96)*, volume 1163 of *Lecture Notes in Computer Science*, pages 244–251. Springer Berlin Heidelberg, 1996.

[2] Elli Androulaki, Seung Geol Choi, Steven M. Bellovin, and Tal Malkin. Reputation systems for anonymous networks. In *Proceedings of the 8th International Symposium on Privacy Enhancing Technologies*, PETS '08, pages 202–218, Berlin, Heidelberg, 2008. Springer-Verlag.

[3] Dan Boneh and Hovav Shacham. Group signatures with verifier-local revocation. In *Proceedings of the 11th ACM Conference on Computer and Communications Security*, CCS '04, pages 168–177, New York, NY, USA, 2004. ACM.

[4] Stefan Brands. Untraceable off-line cash in wallets with observers (extended abstract). In *Proceedings of the 13th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '93, pages 302–318, London, UK, UK, 1994. Springer-Verlag.

[5] David Chaum, Amos Fiat, and Moni Naor. Untraceable electronic cash. In Shafi Goldwasser, editor, *Advances in Cryptology (Crypto '88)*, volume 403 of *Lecture Notes in Computer Science*, pages 319–327. Springer New York, 1990.

[6] David Chaum and Eugène van Heyst. Group signatures. In DonaldW. Davies, editor, *Advances in Cryptology (EUROCRYPT '91)*, volume 547 of *Lecture Notes in Computer Science*, pages 257–265. Springer Berlin Heidelberg, 1991.

[7] H. Chien, J. Jan, and Y. Tseng. RSA-based partially blind signature with low computation. In *Proceedings of the Eighth International Conference on Parallel and Distributed Systems*, ICPADS '01, pages 385–, Washington, DC, USA, 2001. IEEE Computer Society.

[8] Ivan Damgård, Kasper Dupont, and Michael Østergaard Pedersen. Unclonable group identification. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 555–572. Springer Berlin Heidelberg, 2006.

[9] Uriel Feige, Amos Fiat, and Adi Shamir. Zero-knowledge proofs of identity. *Journal of Cryptology*, 1(2):77–94, 1988.

[10] Eiichiro Fujisaki and Koutarou Suzuki. Traceable ring signature. In *Proceedings of the 10th International Conference on Practice and Theory in Public-key Cryptography*, PKC'07, pages 181–200, Berlin, Heidelberg, 2007. Springer-Verlag.

[11] Kevin Hoffman, David Zage, and Cristina Nita-Rotaru. A survey of attack and defense techniques for reputation systems. *ACM Comput. Surv.*, 42(1):1:1–1:31, December 2009.

[12] Alexander Jungmann, Sonja Brangewitz, Ronald Petrlic, and Marie Christin Platenius. Incorporating reputation information into decision-making processes for markets of composed services. *International Journal On Advances in Intelligent Systems*, 7(4), 2014.

[13] Florian Kerschbaum. A verifiable, centralized, coercion-free reputation system. In *Proceedings of the 8th ACM workshop on Privacy in the electronic society*, WPES '09, pages 61–70, New York, NY, USA, 2009. ACM.

[14] Ronald Petrlic, Sascha Lutters, and Christoph Sorge. Privacy-preserving reputation management. In *Symposium on Applied Computing, SAC 2014, Gyeongju, Republic of Korea - March 24 - 28, 2014*, pages 1712–1718, 2014.

[15] Stefan Schiffner, Sebastian Clauß, and Sandra Steinbrecher. Privacy, liveliness and fairness for reputation. In *Proceedings of the 37th international conference on Current trends in theory and practice of computer science*, SOFSEM'11, pages 506–519, Berlin, Heidelberg, 2011. Springer-Verlag.