Anonymous and Publicly Linkable Reputation Systems (Short Paper)

Johannes Blömer*, Jakob Juhnke**, and Christina Kolb*

University of Paderborn,Department of Computer Science,Germany
{bloemer,jakob.juhnke,christina.kolb}@uni-paderborn.de

Abstract. We consider reputation systems where users are allowed to rate products that they purchased previously. To obtain trustworthy reputations, they are allowed to rate these products only once. As long as they do so, the users stay anonymous. Everybody is able to detect users deviating from the rate-products-only-once policy and the anonymity of such dishonest users can be revoked by a system manager. In this paper we present formal models for such reputation systems and their security. Based on group signatures we design an efficient reputation system that meets all our requirements.

Keywords: Reputation, trust, group signatures, anonymity, linkability, verifier-local revocation, traceability, strong-exculpability

1 Introduction

Reputation systems are an increasingly popular tool to give providers and customers valuable information about previous transactions. To provide trustworthy, reliable, and honest ratings there is a need for anonymous reputation systems that also guarantee that customers rate products only once. To further increase trust in the system, everyone - even outsiders - should be able to verify the validity of ratings. In this paper, we propose models for secure and anonymous reputation systems and give an efficient construction of such a system.
Some of the properties for reputation systems stated above have been studied in the context of group signatures, as defined in [3] for the static and in [4] for the dynamic case. However, the concept of group signatures does not meet all the requirements for reputation systems. In particular, reputation systems do not consist of a single group of users. Rather one can think of reputation systems as a family of group signature schemes - one for each product.
Moreover, we may have providers with several products. Hence, when looking at security and anonymity group signature schemes for different products can not be considered in isolation. Finally, known constructions of group signatures do not provide all properties that we need for a secure and anonymous reputation system and do not provide them simultaneously.

* This author was partially supported by the German Research Foundation (DFG) within the Collaborative Research Centre On-The-Fly Computing (SFB 901).
** This author was supported by the International Graduate School "Dynamic Intelligent Systems".

Our Contribution. We define models for secure and anonymous reputation systems and give a first construction of such a system based on group signature schemes. We use the terms rating and message synonymously. Our construction provides anonymity, traceability, strong-exculpability, verifier-local revocation, and public linkability. Anonymity means that signatures of honest users are indistinguishable. Traceability means that it is impossible for any set of colluding users to create ratings that can not be traced back to a user of the system. Strong-exculpability means that nobody can produce signatures on behalf of honest users. A system has local-verifier revocation, if revocation messages only have to be sent to signature verifiers, but not to individual signers. Public linkability requires that anyone can decide whether or not two ratings for the same product were created by the same user, i.e. no secret key is required to link messages. Note that public linkability implies that users can only stay anonymous as long as they rate products just once. As a remark, it is well known how to realize the described properties in the context of group signatures, although not necessarily simultaneously.
Our construction of a reputation system is based on the group signature scheme by Boneh, Boyen, and Shacham [7] (BBS) and the dynamic version of the scheme presented by Delerablée and Pointcheval [11]. These schemes already give us anonymity, traceability, and strong-exculpability. To achieve verifier-local revocation we modify a technique by [25]. With the same technique we achieve public linkability. Note that anonymity of group signatures does not imply anonymity in our reputation system. This is due to the fact that providers control the groups corresponding to several products. Hence, they may combine information for different groups to violate anonymity. To prevent this, we need a system manager that contributes a trustworthy component to each group public key. In Section 2 we present a formal model for reputation systems. The security of our system can be shown in the random oracle model and is based on the same assumptions as the BBS scheme [7]. The formal security model and security proofs of our system are given in the full version of this paper [5].

Related Work. Reputation systems are a popular research topic in economics and computer science, see for example [1,10,12,13,18,19]. Although privacy, i.e. anonymity and security, i.e. unforgeability, have been identified as key properties of reputation systems, no generally accepted privacy and security definitions for reputation systems have emerged. Definitions of anonymity based on differential privacy have been proposed in [10,12,26]. These are restricted to special reputation functions. In [1,20,24] cryptography has been proposed as a methodology to achieve anonymity in reputation systems, albeit without providing detailed definitions. In contrast to this, (anonymous) group signatures have been well studied in cryptography and formal security models exist. Important techniques to design group signature schemes were first described by Ateniese et al. [2]. For the case of static groups formal definitions of security were first given by Bellare, Micciancio and Warinschi [3], for dynamic groups by Bellare, Shi and Zhang [4]. Both works provide frameworks to construct group signature schemes. One of

the most efficient static schemes is that of Boneh, Boyen and Shacham (BBS) [7]. Schemes with verifier-local revocation include [8,25], linkable, though not publicly linkable, group signature schemes include [17,14,23]. In the context of ring signatures different definitions of linkability have been considered before, for example in [15,9,27,22]. Our definition of public linkability is based on the definition given in [15].

2 A Model for Reputation Systems

Our model for reputation systems is based on the model for dynamic group signature schemes by Bellare, Shi, and Zhang [4]. Therefore, we will use the same notation for the authorities, algorithms and security properties as in [4]. From now on the system manager will be called group manager and providers will be called key issuers, because these are their main roles in our reputation system.

Algorithms. A reputation system consists of one authority called the group manager, a set of authorities called the key issuers, and a set of users. The group manager is assumed to be honest, provides the group manager's public key *gmpk* and is able to trace group members. Every key issuer provides *items* with corresponding *item*-based public keys $ipk[item]$, which will be used by the group members to rate/vote a specific *item*. Users have unique identities $i \in \mathbb{N}$ and may become group members by registering at the group manager.
The specification of a reputation system is a tuple of polynomial-time algorithms $\mathcal{RS} = (\text{KeyGen}_{GM}, \text{KeyGen}_{KI}, \text{KeyGen}_{U}, \text{Register}_{GM}, \text{Register}_{U}, \text{Join}, \text{Issue}, \text{Revoke}, \text{Sign}, \text{Verify}, \text{Link}, \text{Open})$. Their functionality is described as follows.
$\mathbf{KeyGen_{GM}()}$: This randomized algorithm is run in the setup phase by the group manager to create the public and secret key pair $(gmpk, gmsk)$. The secret key *gmsk* contains elements which allow tracing of group members and the creation of revocation tokens.
$\mathbf{KeyGen_{KI}(item)}$: This randomized algorithm is run by a key issuer for every $item \in \{0,1\}^*$ he provides to obtain an *item*-based public and secret key pair $(ipk[item], isk[item])$. The tuple $(item, ipk[item])$ is added to the public *ItemList*.
$\mathbf{KeyGen_{U}(i)}$: This randomized algorithm is run to create the user's public and secret key pair $(upk[i], usk[i])$. The user's public key $upk[i]$ is used during the registration to the group, the corresponding secret key $usk[i]$ is used to create signatures.
$\mathbf{Register_{GM}(St_{GM}, M_{GM}), Register_{U}(St_{U}, M_{U})}$: These randomized interactive algorithms are run by the group manager and a user $i \in \mathbb{N}$, who wants to become a group member. If the group manager accepts, the tuple $(i, upk[i])$ is added to the registration table *reg*. The input parameters of the algorithms are some state information and a message, which was received from the communicating partner. It is assumed that the user starts the interaction.
$\mathbf{Join(St_{U}, M_{U}), Issue(St_{KI}, M_{KI})}$: These randomized interactive algorithms are run by a user $i \in \mathbb{N}$ and a key issuer. The input parameters of the algorithms

are some state information and a message, which was received from the communicating partner. It is assumed that the user starts the interaction. The first message of the user i must contain $upk[i]$ and an *item*. If Issue accepts, the key issuer sends a personal signing key for the given *item* $gsk[i, item]$ to the user and saves the tuple $(upk[i], gsk[i, item])$ in the identification list IL_{item}.

Revoke($gmpk, gmsk, i$): This deterministic algorithm is run by the group manager to revoke signers in case of misuse. Revoke computes the revocation token $grt[i]$ of user i and adds it to the public revocation list $\mathcal{RL}$.

Sign($item, gmpk, ipk[item], gsk[i, item], usk[i], M$): This randomized algorithm is run by users to create signatures for specific *item*s. Given the necessary keys and a message M, Sign computes and outputs a signature σ on M under the given keys.

Verify($item, gmpk, ipk[item], \mathcal{RL}, M, \sigma$): This deterministic algorithm can be run by any user, even by an outsider, to obtain a bit v. We say that σ is a *valid* signature of M with respect to the given keys, iff the bit v is 1.

Link($item, gmpk, ipk[item], (M', \sigma'), (M'', \sigma'')$): This deterministic algorithm can be run by any user, even by an outsider, to obtain a bit ℓ. We call σ' and σ'' *publicly linkable* signatures, iff the bit ℓ is 1.

Open($gmpk, gmsk, M, \sigma$): This deterministic algorithm is run by the group manager to *open* signatures. Using $gmsk$, Open outputs the identity of the signer of σ or `failure`.

Figure 1 illustrates the interaction of the described parties and the algorithms involved. It is not hard to see that the number of key issuers is not important in this model: a single key issuer has the same capabilities as a colluding set of key issuers. Therefore, in all formal definitions we will only consider the case that the number of key issuers is 1. Additionally, we assume that the signing keys from the key issuer given to a user are publicly verifiable, i.e. the correctness of keys can be checked using only public parameters.

Correctness. Informally, a reputation system must satisfy the following correctness requirements:

1. honestly created signatures of non-revoked users will be accepted by the Verify algorithm,
2. honestly created signatures can be traced back to the correct signer,
3. two different signatures for the same *item* created by a single user will be detected by the Link algorithm.

Security Notions. To model the different attack capabilities of an adversary, we introduce oracles, which will be used in the definitions of security. We present only informal descriptions of these oracles, their formal definitions are given in the full version of this paper [5] and are based on [4] and [14]. We assume that a security experiment has run $\text{KeyGen}_{GM}()$ to obtain $(gmpk, gmsk)$, and manages the global sets $\mathcal{HU}$, $\mathcal{CU}$, $\mathcal{RU}$, $\mathcal{JIU}$, $\mathcal{GS}$, *reg* and *ItemList*. Except *ItemList* and *reg* all sets are only used within the formal definitions of the oracles and the

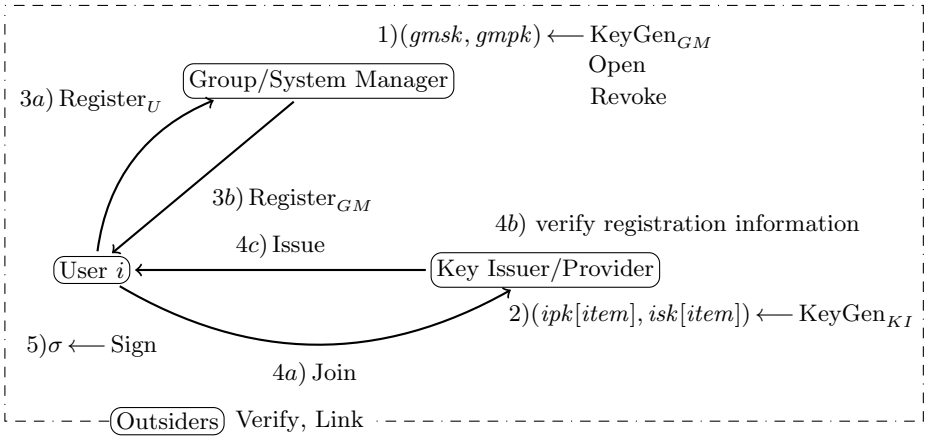

Fig. 1. Interaction of the parties within a reputation system.

security experiments. By $\mathcal{HU}$ we denote the set of honest users, by $\mathcal{CU}$ the set of corrupted users. The set $\mathcal{RU}$ contains all identities of users that currently engage in the registration protocol. The set $\mathcal{JIU}$ contains all identities of users that currently engage in the Join-Issue protocol. By $\mathcal{GS}$ we denote the set of queried signatures. All sets are assumed to be initially empty.

AddU(i): To add honest users to the group, the adversary can call this *add user* oracle. The oracle adds i to $\mathcal{HU}$ and executes the registration protocol by running Register_{GM} and Register_{U}. The oracle returns $upk[i]$ to the adversary.

AddItem($item$): An adversary can add *item*s by using this *add item* oracle. The oracle then runs the KeyGen_{KI} algorithm and returns $ipk[item]$ to the adversary.

USK(i): To get the secret key $usk[i]$ of an honest user i, an adversary can call the *user secret key* oracle. Then the user i is added to $\mathcal{CU}$.

GSK(i, upk, $item$): To get the secret signing key $gsk[i, item]$ of user i for a specified *item*, an adversary can call the *signing key* oracle.

RevU(i): To get the *revocation token* of user i, an adversary can call the *revoke user* oracle. The oracle runs the Revoke algorithm and returns $grt[i]$ to the adversary.

GSig(i, upk, $item$, M): An adversary can use the *signing* oracle to obtain a valid signature for the message M with respect to the signing key of user i and the *item*-based public key $ipk[item]$. The queried signature is added to $\mathcal{GS}$.

SndToKI(i, $item$, $upk[i]$, M_{KI}): After corruption of user i, the adversary can use the *send to key issuer* oracle to engage in a join protocol with the key issuer. The oracle honestly runs the Issue algorithm and computes a response to M_{KI}.

SndToGM(i, M_{GM}): The *send to group manager* oracle can be used by an adversary to engage in a registration protocol with the honest group manager. The oracle honestly runs the Register_{GM} algorithm and adds the user i to $\mathcal{CU}$.

WItemList($item, ipk$): An adversary can use the *write to item list* oracle to manipulate the *item* based public key of the specified *item*. If $ipk = \varepsilon$ the *item* is deleted from the list. Otherwise, the specified public key is set.
WIdentList($item, i, upk[i], gsk$): Using the *write to identification list* oracle an adversary can modify the secret signing keys of user $i \in \mathbb{N}$ for the specified *item*. If $gsk = \varepsilon$ the key information about user i is deleted from the list.
Open($item, M, \sigma$): The *opening* oracle can be used by the adversary to get the output of the Open algorithm, as long as σ was not produced by the GSig oracle.
In our reputation system we need anonymity, public linkability, traceability, and strong-exculpability. The anonymity and traceability experiments are based on [4], the public linkability experiment is based on [15] and the strong-exculpability experiment is based on [21,2,4]. Complete formal definitions of the oracles and the experiments are given in the full version of the paper [5].
The anonymity experiment $\mathrm{Exp}^{\mathrm{anon-b}}_{\mathcal{A},\mathcal{RS}}(k)$ asks an adversary to distinguish which of two group members signed a message for some *item*, where the identities, the message, and an *item* are chosen by the adversary. The adversary's attack capabilities are strong: it is possible to corrupt the key issuer and all but two users. These two users must be honest because otherwise the adversary could possibly link different signatures or use the revocation token of the users to determine their identities.
The public linkability experiment $\mathrm{Exp}^{\mathrm{publink}}_{\mathcal{A},\mathcal{RS}}(k)$ asks an adversary to output message-signature pairs for a single *item* chosen by the adversary, such that all pairs are valid and there are no two pairs that can be linked. The number of pairs must be one more than the number of users in the group. We allow the adversary to corrupt all users, but the key issuer has to be honest.
The traceability experiment $\mathrm{Exp}^{\mathrm{trace}}_{\mathcal{A},\mathcal{RS}}(k)$ asks an adversary to output a message-signature pair, for some *item* chosen by the adversary, which is valid but can not be traced back to a corrupted user. In this experiment the key issuer is assumed to be honest.
The strong-exculpability experiment $\mathrm{Exp}^{\mathrm{str-ex}}_{\mathcal{A},\mathcal{RS}}(k)$ asks an adversary to output a message-signature pair, for some *item* chosen by the adversary, which is valid and can be traced back to an honest user. We give an adversary the possibility to corrupt users and the key issuer. Because the key issuer can always generate signing keys for non-existing users, we force the adversary to output a signature on behalf of an honest user.

Discussion: The described experiments imply two different attack scenarios:

> In the first scenario, for anonymity and strong-exculpability, we allow an adversary to corrupt key issuers and users. One could argue, that there is an oracle missing to allow an adversary to send corrupted data to honest users in the Join-Issue protocol. But this functionality is covered by the SndToGM, WItemList, and WIdentList oracles and by publicly verifiable signing keys.
> In the second scenario, for public linkability and traceability, key issuers are assumed to be honest, whereas users can be corrupted. In particular,

this implies that users and key issuers are disjoint sets. The restriction to honest key issuers is necessary because a corrupted key issuer could generate secret keys for non-existing users. With an appropriate identity management this can be prevented and we could also allow corrupted key issuers in the experiments for public linkability and traceability.

An important issue is that of timing the operations. The key issuer may correlate transactions and ratings by their timing, thereby threatening the anonymity of users. Hence, our reputation systems needs a mechanism to prevent such attacks. In [10], [20], and [16] different solutions to this problem are proposed, which can be incorporated into our construction.

3 Our Construction

In this section we describe our reputation system by giving formal definitions of all algorithms stated in Section 2. The reputation system is based on the group signature schemes [7], [11] and [25]. An intuition for our system can be obtained from the honest-verifier zero-knowledge proof of knowledge for the so-called extended q-SDH problem explained in the full version of this paper [5]. We assume the communication between users and the group manager and between users and the key issuer to take place via secure channels. Furthermore, the user's public key $upk[i]$ is certified by the group manager, such that the key issuer can verify the integrity of the public keys during the Join-Issue protocol. In the following definitions we consider bilinear groups $\mathbb{G}_1$ and $\mathbb{G}_2$, and two hash functions modeled as random oracles: $H\colon \{0,1\}^* \longrightarrow \mathbb{Z}_p$ and $H_1\colon \{0,1\}^* \longrightarrow \mathbb{G}_2$. Furthermore, as in [7], we use Linear Encryption - a CPA-secure Elgamal-like encryption scheme based on the Decision Linear Diffie-Hellman Assumption.

$\text{KeyGen}_{GM}()$:

1. Select $w \xleftarrow{\$} \mathbb{G}_1$, $\hat{d} \xleftarrow{\$} \mathbb{G}_2$, $\xi_1, \xi_2, \zeta \xleftarrow{\$} \mathbb{Z}_p$ and compute $u := w^{\frac{1}{\xi_1}}$, $v := w^{\frac{1}{\xi_2}}$, $d := \psi(\hat{d})$, $h := d^{\zeta}$. The values (u, v, w) are the public key of the Linear Encryption, the values (ξ_1, ξ_2) are the corresponding secret key, $\hat{d}, d$ and h are the basis for public linkability and revocation.
2. Set $gmpk := (u, v, w, h, d, \hat{d})$ and $gmsk := (\xi_1, \xi_2, \zeta)$.

$\text{KeyGen}_{KI}(item)$:

1. Select $g_{2_{item}} \xleftarrow{\$} \mathbb{G}_2$, $\gamma_{item} \xleftarrow{\$} \mathbb{Z}_p$, set $g_{1_{item}} := \psi(g_{2_{item}})$, $W_{item} := g_{2_{item}}^{\gamma_{item}}$.
2. Set $ipk[item] := (g_{1_{item}}, g_{2_{item}}, W_{item})$, add it to the *ItemList* and keep $isk[item] := \gamma_{item}$ secret.

$\text{KeyGen}_{U}(i)$:

1. Select $y_i \xleftarrow{\$} \mathbb{Z}_p$, set $upk[i] := h^{y_i}$ and $usk[i] := y_i$.

$\text{Register}_{GM}(\text{St}_{GM}, M_{GM}), \text{Register}_{U}(\text{St}_{U}, M_{U})$:

1. The user sends his identity i to the group manager.

2. If $reg[i] = \varepsilon$, the group manager runs KeyGen_U to obtain the tuple $(upk[i], usk[i])$, sets $reg[i] := (i, upk[i])$ and sends $(upk[i], usk[i])$ to the user i.

$\text{Join}(\text{St}_U, M_U), \text{Issue}(\text{St}_{KI}, M_{KI})$:

1. The user looks up $ipk[item] = (g_{1_{item}}, g_{2_{item}}, W_{item})$ in the *ItemList* and sends $(i, upk[i])$ to the key issuer.
2. The key issuer checks that there is no entry $(upk[i], \cdot)$ in the identification list IL_{item}, selects $x_{i_{item}} \xleftarrow{\$} \mathbb{Z}_p$, computes $A_{i_{item}} := (g_{1_{item}} \cdot upk[i])^{\frac{1}{x_{i_{item}} + \gamma_{item}}}$, gives $gsk[i, item] := (A_{i_{item}}, x_{i_{item}})$ to user i, and saves $(upk[i], gsk[i, item])$ in IL_{item}.

$\text{Revoke}(gmpk, gmsk, i)$:

1. Look up $upk[i]$ in $reg[i]$ and compute $D_i := upk[i]^{\frac{1}{\zeta}} = (h^{y_i})^{\frac{1}{\zeta}} = d^{y_i}$ using $gmsk$ and add the revocation token $grt[i] := D_i$ to the revocation list $\mathcal{RL}$.

$\text{Sign}(item, gmpk, ipk[item], gsk[i, item], usk[i], M)$:

1. Obtain the value $\hat{f} \in \mathbb{G}_2$ by $\hat{f} := H_1(item)$, choose $\alpha, \beta, \mu \xleftarrow{\$} \mathbb{Z}_p$ and compute $T_1 := u^{\alpha}$, $T_2 := v^{\beta}$, $T_3 := A_{i_{item}} \cdot w^{\alpha+\beta}$, $T_4 := d^{\mu}$, $T_5 := \psi(\hat{f})^{\mu+y_i}$ and the helper values $\delta_1 := \alpha \cdot x_{i_{item}}$ and $\delta_2 := \beta \cdot x_{i_{item}}$.
2. Select $r_\alpha, r_\beta, r_x, r_y, r_\mu, r_{\delta_1}, r_{\delta_2} \xleftarrow{\$} \mathbb{Z}_p$ and compute $R_1 := u^{r_\alpha}$, $R_2 := v^{r_\beta}$, $R_3 := \text{e}(T_3, g_{2_{item}})^{r_x} \cdot \text{e}(w, W_{item})^{-r_\alpha - r_\beta} \cdot \text{e}(w, g_{2_{item}})^{-r_{\delta_1} - r_{\delta_2}} \cdot \text{e}(h, g_{2_{item}})^{-r_y}$ $R_4 := T_1^{r_x} \cdot u^{-r_{\delta_1}}$, $R_5 := T_2^{r_x} \cdot v^{-r_{\delta_2}}$, $R_6 := d^{r_\mu}$, $R_7 := \psi(\hat{f})^{r_\mu + r_y}$.
3. Compute $c := H(M,\ item,\ T_1,\ T_2,\ T_3,\ T_4,\ T_5,\ R_1,\ R_2,\ R_3,\ R_4,\ R_5,\ R_6,\ R_7)$ and $s_\alpha := r_\alpha + c \cdot \alpha$, $s_\beta := r_\beta + c \cdot \beta$, $s_x := r_x + c \cdot x_{i_{item}}$, $s_y := r_y + c \cdot y_i$, $s_\mu := r_\mu + c \cdot \mu$, $s_{\delta_1} := r_{\delta_1} + c \cdot \delta_1$, $s_{\delta_2} := r_{\delta_2} + c \cdot \delta_2$.
4. Output $\sigma := (item,\ T_1,\ T_2,\ T_3,\ T_4,\ T_5,\ c,\ s_\alpha,\ s_\beta,\ s_x,\ s_y,\ s_\mu,\ s_{\delta_1},\ s_{\delta_2})$.

$\text{Verify}(item, gmpk, ipk[item], \mathcal{RL}, M, \sigma)$:

1. Obtain the value $\hat{f} \in \mathbb{G}_2$ by $\hat{f} := H_1(item)$ and compute the values $R_1 := u^{s_\alpha} \cdot T_1^{-c}$, $R_2 := v^{s_\beta} \cdot T_2^{-c}$,

$$R_3 := \frac{\text{e}(T_3, g_{2_{item}})^{s_x} \cdot \text{e}(w, W_{item})^{-s_\alpha - s_\beta} \cdot \text{e}(w, g_{2_{item}})^{-s_{\delta_1} - s_{\delta_2}}}{\text{e}(T_3, W_{item})^{c} \cdot \text{e}(g_1, g_{2_{item}})^{-c} \cdot \text{e}(h, g_{2_{item}})^{s_y}},$$

$R_4 := T_1^{s_x} \cdot u^{-s_{\delta_1}}$, $R_5 := T_2^{s_x} \cdot v^{-s_{\delta_2}}$, $R_6 := d^{s_\mu} \cdot T_4^{-c}$, $R_7 := \psi(\hat{f})^{s_\mu + s_y} \cdot T_5^{-c}$.

2. Check that $c \stackrel{?}{=} H(M, item, T_1, T_2, T_3, T_4, T_5, R_1, R_2, R_3, R_4, R_5, R_6, R_7)$. If this holds, then accept, otherwise reject.
3. For each element $D \in \mathcal{RL}$ check whether D is encoded in (T_4, T_5): $\text{e}\left(T_5, \hat{d}\right) \stackrel{?}{=} \text{e}(D \cdot T_4, \hat{f})$. If this is false for all $D \in \mathcal{RL}$, then the signer of σ has not been revoked and Sign accepts, otherwise rejects.
4. If both checks accept, then output 1, otherwise 0.

$\text{Link}(item, gmpk, ipk[item], (M', \sigma'), (M'', \sigma''))$:

1. Verify the signatures σ' and σ'' and compute the value $\hat{f} := H_1(item)$.

2. Output 1, iff σ' and σ'' are valid and $\mathrm{e}\left(\frac{T_5'}{T_5''}, \hat{d}\right) \stackrel{?}{=} \mathrm{e}\left(\frac{T_4'}{T_4''}, \hat{f}\right)$ holds.

Open($gmpk, gmsk, M, \sigma$):

1. Check that σ is a valid signature. If not, output `failure`.
2. Compute $A_{i_{item}} := T_3 \cdot T_1^{-\xi_1} \cdot T_2^{-\xi_2}$ using $gmsk$ and look up the user index i from the identification list IL_{item}.
3. If no entry for $A_{i_{item}}$ can be found in IL_{item} return `failure`, otherwise return i.

Theorem 1. *The above reputation system is* correct. *Furthermore, assuming the q-SDH Problem is hard in the bilinear groups* $(\mathbb{G}_1, \mathbb{G}_2)$ *and the Decision Linear Problem is hard in* $\mathbb{G}_1$*, the reputation system is* anonymous, publicly linkable, traceable*, and* strongly exculpable.

The q-SDH Problem and the Decision Linear Problem are standard problems in pairing-based cryptography and formal definitions can be found in [7]. Both problems are hard to solve in the Generic Group Model [6,7].
Formal definitions of the security properties and proofs of security will be given in the full version of this paper [5].

References

1. Elli Androulaki, SeungGeol Choi, Steven M. Bellovin, and Tal Malkin. Reputation Systems for Anonymous Networks. In *Privacy Enhancing Technologies*, volume 5134 of *LNCS*, pages 202–218. Springer, 2008.
2. Giuseppe Ateniese, Jan Camenisch, Marc Joye, and Gene Tsudik. A Practical and Provably Secure Coalition-Resistant Group Signature Scheme. In *CRYPTO 2000*, volume 1880 of *LNCS*, pages 255–270. Springer, 2000.
3. Mihir Bellare, Daniele Micciancio, and Bogdan Warinschi. Foundations of Group Signatures: Formal Definitions, Simplified Requirements, and a Construction Based on General Assumptions. In *EUROCRYPT 2003*, pages 614–629. Springer, 2003.
4. Mihir Bellare, Haixia Shi, and Chong Zhang. Foundations of Group Signatures: The Case of Dynamic Groups. In *CT-RSA 2005*, volume 3376 of *LNCS*, pages 136–153. Springer, 2005.
5. Johannes Blömer, Jakob Juhnke, and Christina Kolb. Anonymous and Publicly Linkable Reputation Systems, 2014.
6. Dan Boneh and Xavier Boyen. Short Signatures Without Random Oracles. In *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 56–73. Springer, 2004.
7. Dan Boneh, Xavier Boyen, and Hovav Shacham. Short Group Signatures. In *CRYPTO 2004*, volume 3152 of *LNCS*, pages 41–55. Springer, 2004.
8. Dan Boneh and Hovav Shacham. Group Signatures with Verifier-local Revocation. In *CCS 2004*, pages 168–177. ACM, 2004.
9. Sherman S.M. Chow, Willy Susilo, and Tsz Hon Yuen. Escrowed Linkability of Ring Signatures and Its Applications. In *VIETCRYPT 2006*, volume 4341 of *LNCS*, pages 175–192. Springer, 2006.
10. Sebastian Clauß, Stefan Schiffner, and Florian Kerschbaum. k-anonymous Reputation. In *ASIA CCS 2013*, pages 359–368. ACM, 2013.

11. Cécile Delerablée and David Pointcheval. Dynamic Fully Anonymous Short Group Signatures. In *VIETCRYPT 2006*, volume 4341 of *LNCS*, pages 193–210. Springer, 2006.
12. Chrysanthos Dellarocas. Immunizing Online Reputation Reporting Systems Against Unfair Ratings and Discriminatory Behavior. In *EC 2000*, pages 150–157. ACM, 2000.
13. Roger Dingledine, Nick Mathewson, and Paul Syverson. Reputation in P2P Anonymity Systems. In *Workshop on Economics of Peer-to-Peer Systems*, volume 92, 2003.
14. Matthew Franklin and Haibin Zhang. Unique Group Signatures. In *ESORICS 2012*, volume 7459 of *LNCS*, pages 643–660. Springer, 2012.
15. Eiichiro Fujisaki and Koutarou Suzuki. Traceable Ring Signature. In *PKC 2007*, volume 4450 of *LNCS*, pages 181–200. Springer, 2007.
16. Micheal T. Goodrich and Florian Kerschbaum. Privacy-Enhanced Reputation-Feedback Methods to Reduce Feedback Extortion in Online Auctions. In *CODASPY 2011*, pages 273–282. ACM, 2011.
17. Jung Yeon Hwang, Sokjoon Lee, Byung-Ho Chung, Hyun Sook Cho, and DaeHun Nyang. Group signatures with controllable linkability for dynamic membership. *Information Sciences*, 222:761–778, 2013.
18. Audun Jøsang and Roslan Ismail. The Beta Reputation System. In *BLED 2002*, pages 41–55, 2002.
19. Sepandar D. Kamvar, Mario T. Schlosser, and Hector Garcia-Molina. The EigenTrust Algorithm for Reputation Management in P2P Networks. In *WWW 2003*, pages 640–651. ACM, 2003.
20. Florian Kerschbaum. A Verifiable, Centralized, Coercion-free Reputation System. In *WPES 2009*, pages 61–70. ACM, 2009.
21. Aggelos Kiayias and Moti Yung. Group Signatures: Provable Security, Efficient Constructions and Anonymity from Trapdoor-Holders, 2004.
22. Joseph K. Liu, Victor K. Wei, and Duncan S. Wong. Linkable Spontaneous Anonymous Group Signature for Ad Hoc Groups. In *ACISP 2004*, volume 3108 of *LNCS*, pages 325–335. Springer, 2004.
23. Mark Manulis, Ahmad-Reza Sadeghi, and Jörg Schwenk. Linkable Democratic Group Signatures. In *ISPEC 2006*, volume 3903 of *LNCS*, pages 187–201. Springer, 2006.
24. Antonis Michalas and Nikos Komninos. The Lord of the Sense: A Privacy Preserving Reputation System for Participatory Sensing Applications. In *Computers and Communication*, volume 23 of *ISCC*, pages 1–6. IEEE, 2014.
25. Toru Nakanishi and Nobuo Funabiki. A Short Verifier-Local Revocation Group Signature Scheme with Backward Unlinkability. In *Advances in Information and Computer Security*, volume 4266 of *LNCS*, pages 17–32. Springer, 2006.
26. Sandra Steinbrecher. Design Options for Privacy-Respecting Reputation Systems within Centralised Internet Communities. In *Security and Privacy in Dynamic Environments*, volume 201 of *IFIP*, pages 123–134. Springer, 2006.
27. Patrick P. Tsang and Victor K. Wei. Short Linkable Ring Signatures for E-voting, E-cash and Attestation. In *IPSEC 2005*, volume 3439 of *LNCS*, pages 48–60. Springer, 2005.