Reputation Schemes for Pervasive Social Networks with Anonymity

Lydia Garms* Keith Martin† Siaw–Lynn Ng‡
Information Security Group, Royal Holloway, University of London
Email: *Lydia.Garms.2015@rhul.ac.uk, †Keith.Martin@rhul.ac.uk, ‡S.Ng.rhul.ac.uk

September 26, 2017

Abstract

Pervasive Social Networks of Strangers are available constantly, connecting users directly with no preexisting relationships. Such networks demand a high level of privacy, as users should be anonymous and their behaviour unlinkable. However, the anonymity of users could lead to abusive messages, spam, or fraud. A possible solution is to keep track of users' reputations, based on feedback from other users. However, the need to collate feedback on user behaviour to form reputations is at odds with the need for privacy. Anonymous and unlinkable feedback is also required, whilst multiple feedbacks given on the same item should be detected. To resolve this, we use group signatures and direct anonymous attestation, to give a reputation-based messaging scheme.

1 Introduction

1.1 Pervasive Social Networks

A Pervasive Social Network (PSN) [14] is a social network that is available constantly. The term originates from pervasive technology, such as smart phones and wearables.

Communication in PSNs takes place on the go. Therefore, users should be able to communicate directly, without the involvement of a server to verify that messages satisfy the requirements for the social network, such as checking that the sender is enrolled in the scheme, or that messages follow a prescribed format. Users who do not conform may be revoked from the network.

1.2 Pervasive Social Networks of Strangers

Pervasive Social Networks of Strangers are PSNs where users communicate with those they have no previous relationship with. Examples of this include the "meeting new friends" feature of the app Kik [2], which had 300 million registered users in May 2016 [3], and Firechat [1], on which users broadcast messages publicly.

In PSNs of Strangers, users are talking to people they do not trust. Therefore, broadcast public messages should be untraceable, which means we cannot discover a user's long term identity. For example, FireChat was used during the 2014 protests in Hong Kong for anonymous communication [1]. Also, users may not want their behaviour on the social network to be linked together, as one slip could de-anonymise their entire behaviour.

1.3 Reputation Schemes for PSNs of Strangers

With no preexisting trust relationships between users, and no ability to build user profiles, it is not clear how users could judge the accuracy of public messages they receive. Therefore, reputation schemes, which

*This author was supported in this research by the EPSRC and the UK government as part of the Centre for Doctoral Training in Cyber Security at Royal Holloway, University of London (EP/K035584/1)

keep track of a reputation value for each user, could help by replacing preexisting trust. This represents how well a user has behaved in the past, which gives an indication of how that user may be behaving currently. Users that behave maliciously can be punished by revoking them.

To form reputation values, users give feedback on other users' public messages. In order to collate this feedback, the subject of the feedback must be linked to an identity. Multiple feedback on the same subject should also be detected as this unfairly skews users' reputations.

The incorporation of a reputation scheme must leave privacy intact. Unfortunately the two are at odds with each other. Users' behaviour must be linked to form a reputation, but users' behaviour should be unlinkable to protect their privacy. This is the problem we aim to solve.

1.4 Existing Work

Existing reputation schemes for social networks can be categorized into distributed and centralised schemes.

Distributed reputation schemes, such as [9], have no central server. These schemes use local reputation, evaluated by a user on other users, based on first hand experience or second hand accounts [12]. However, to form local reputations, a user's behaviour must be linkable to other users. Therefore, distributed reputation schemes cannot have unlinkability as a privacy guarantee.

In a centralised scheme, privacy can be achieved simply. The server can authenticate the user, discover the correct reputation, and then distribute the message with a fresh anonymous pseudonym. An example of a centralised reputation scheme is AnonRep [15] for internet message boards. However, in our setting, users must be able to communicate directly.

A hybrid is necessary, with a central server to collate feedback for privacy, while users communicate directly.

An example of a hybrid scheme is PerChatRep [14]. Local reputations are used, as well as a global reputation calculated by a Trusted Server (TS). Because privacy is achieved by changing the pseudonyms of users regularly, it does not achieve total unlinkability. If pseudonyms change for each new message, then the receiver must receive new pseudonyms from the TS each time. We would like all users' behaviour to be unlinkable, without constant communication with the server.

Another example of a hybrid reputation scheme is devised in [13] for Vehicular Ad-hoc Networks (VANETs). In this scheme messages and feedback are anonymous and unlinkable to other users. Constant communication with a server is not required. However, for PSNs of Strangers it is important that feedback is anonymous and unlinkable even to the TS, to encourage honest feedback.

Therefore, we propose a reputation–based messaging scheme that fully satisfies our requirements.

1.5 Our Contribution

We use group signature schemes and direct anonymous attestation to create a scheme that enables users to broadcast public messages anonymously alongside reputation, without a server. Public messages are unlinkable and untraceable to users. Feedback is anonymous and unlinkable to all, unless multiple feedback is given on the same subject, which can be detected.

Section 2 will specify the functionality and privacy required. Section 3 describes the signature schemes that will be used in Section 4 to construct the reputation-based messaging scheme. Section 5 evaluates whether the requirements from Section 2 are satisfied.

2 Requirements for our Reputation-based Messaging Scheme

Our scheme consists of a set of users, and a Trusted Server (TS). Each user should be able to broadcast public messages. Feedback can be given to the TS on these public messages. The requirements are as follows.

1. Verifiability

   (a) Only users who are enrolled in the scheme should be able to send public messages, or feedback.

(b) Users should be able to verify the reputation of the author of public messages.

(c) The TS should be able to verify the reputation of the author of feedback.

2. Correctness

   (a) The TS should be able to link public messages by author, so that feedback can be attributed correctly to build reputations.

   (b) The TS should be able to determine whether multiple feedback has been given on the same subject, so that reputations formed are fair.

   (c) It should be possible to revoke users who behave badly from the scheme. A revoked user is unable to broadcast public messages or give feedback.

3. Privacy

   (a) Public messages are untraceable and unlinkable by other users.

   (b) If feedback is given once per public message, feedback is untraceable and unlinkable to all.

3 Signature Schemes

3.1 Fulfilling the Requirements

We first consider broadcasting public messages. For a user to prove they have enrolled, as in requirement 1(a), they must send a signature. These signatures must be untraceable and unlinkable due to 3(a). For 2(a), the TS must be able to link messages with the same author. To allow this we could give the TS a secret key. Group Signature Schemes [8] satisfy this.

When sending feedback, for 1(a), a user must prove they have enrolled. There should be no opener, because feedback should be untraceable and unlinkable to all for 3(b). However, for 2(b) it must be possible to only link users who give multiple feedback on the same item. Direct Anonymous Attestation (DAA) [6] satisfies this as it allows messages with the same author and "basename" to be linked. When the basename is the subject of the feedback, we can link multiple feedback on a message.

We can satisfy 1(b) and 1(c), by binding reputation to both types of signatures. We use BBS* [13], to do this for sending public messages. We introduce a modified DAA scheme CDL*, to do this for providing feedback.

In BBS* and CDL* users must regularly update their secret keys. We can revoke users, for 2(c), by no longer updating them. This allows us to avoid revocation lists.

3.2 Group Signatures

Group signatures [8] prove the signer is a member of a group, without revealing their identity. A secret key can open the signature to reveal the signer.

3.3 BBS*

BBS* [13] is a modification of BBS [5] to allow the visible binding of a reputation value to the signature, and to allow updates of a user's secret key following reputation change without a secure channel.

Unfortunately, in [13] the user can manipulate their secret key to increase their reputation by an arbitrary factor, by raising the RCT_i from the original scheme to the power of this factor. We solve this by hashing ${K_i}^r$ in BUpdate* as shown below. We give a brief description of each algorithm, details can be found in [13].

- BKeyGen*: on input the total number of time intervals l, the TS outputs the public parameters $gpp^m = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, p, \psi, \hat{t}_1, H_1, H_2)$, $\mathcal{K} = \{K_i : i \in [l]\}$, the public key gpk^m for sending messages and the TS's secret key smk.

- BJoin*: on input the secret message key smk, the TS enrols a user b and generates their group member secret key $gsk^m_b = (Z_b, x_b)$.
- BUpdate*: this protocol, between a user b and the TS, updates user b's secret key $gsk^m_b = (Z_b, x_b)$ so it is bound to their new reputation $r_{b,i}$ at time interval $\mathbb{T}_i$. The $smk = (\gamma, \eta_1, \eta_2)$, $K_i \in \mathcal{K}$, $r_{b,i}$ and gsk^m_b are used to compute $RCT_i = H_1(K_i^{r_{b,i}})^{\frac{1}{\gamma+x_b}}$, which is sent publicly to the user b, alongside $r_{b,i}$, and used to output $gsk^m_{b,i} = (Z_{b,i}, x_b)$. The hash function H_1 is introduced to stop the above attack.
- BSign*: the user b, with reputation $r_{b,i}$, signs message M during time interval $\mathbb{T}_i$ using the updated public key $gpk^m_{i,r_{b,i}}$ and their secret key $gsk^m_{b,i}$ to output a signature σ, which is bound to $r_{b,i}$.
- BVerify*: verifies a signature σ including reputation r on M. $K_i \in \mathcal{K}$ and r is used to compute the updated public key $gpk^m_{i,r}$ from gpk^m, which is used to verify the signature.
- BOpen*: the TS uses the secret message key smk and public key gpk^m to discover $Z_{b,i}$, the identity of the author b of the signature σ on message M.

3.4 Direct Anonymous Attestation

Direct Anonymous Attestation (DAA) [6] is a signature scheme that allows a user to authenticate themselves as an authorised signer, without revealing their identity. Unlike group signatures, there is no opening function. Instead there is user-controlled linkability – a user can control whether two messages can be linked.

3.5 CDL*

The DAA scheme we will modify [7], is referred to as the CDL scheme. This modification, in CDL*, binds the reputation value to the signatures visibly, using an updated secret key received during CDLUpdate*. This forces users to reveal their reputation when signing.

This modification includes the addition of CDLUpdate*, and small changes in CDLKeyGen*, CDLSign*, CDLVerify* given in Figure 2. The CDL* signature scheme consists of the following algorithms/ protocols.

- CDLKeyGen*: on input the total number of time intervals l, and the set of reputation values $\mathcal{R}$, the TS outputs the public parameters gpp^{fb}, the public set $\mathcal{E}$ and private set Ξ, the public key for feedback gpk^{fb}, and the secret feedback key $sfbk$.
- CDLJoin*: on input the secret feedback key $sfbk$ the TS enrols a user b. The TS outputs the user's credential cre_b for use in CDLUpdate* and the user b outputs their cre_b and their group member secret key for feedback f_b which is not known by the TS.
- CDLUpdate*: this protocol, between a user b and the TS, updates user b's credential cre_b so it is bound to their new reputation $r_{b,i}$ at time interval $\mathbb{T}_i$. The parameters $\xi_{i,r_{b,i}}$, which are known only to the TS, the secret feedback key $sfbk$, cre_b and $r_{b,i}$ are used to compute RCT'_i and RCT''_i, which are sent publicly to the user b, alongside $r_{b,i}$, and used to update the credential to output $cre_{b,i}$.
- CDLSign*: user b, with reputation $r_{b,i}$, signs a feedback report $fbr = (msg, fbk)$ during time interval $\mathbb{T}_i$ using the updated public key $gpk^{fb}_{i,r_{b,i}}$, their secret key for feedback gsk^{fb}_b, and their updated credential $cre_{b,i}$ to output a signature σ, which is bound to $r_{b,i}$.
- CDLVerify*: verifies a signature σ, from time interval $\mathbb{T}_i$, including reputation r, on M. $E_{i,r} \in \mathcal{E}$ and r is used to compute $gpk^{fb}_{i,r}$ from gpk^{fb}, which is used to verify the signature.
- CDLLink*: outputs true only if two valid signatures have the same author and $msg_0 = msg_1$.

```
CDLKeyGen*($l, \mathcal{R}$)
  $(gpp^{fb} = (\mathbb{G}_4, \mathbb{G}_5, \mathbb{G}_6, p, \hat{t}_2, P_4, P_5, H_3, H_4), gpk^{fb},\ sfbk) \leftarrow$ CDLKeyGen($l$)
  $\forall i \in [l], r \in \mathcal{R} \quad \xi_{i,r} \leftarrow\$\, \mathbb{Z}_p^*, E_{i,r} \leftarrow {P_5}^{\xi_{i,r}}$
  $\Xi \leftarrow \{\xi_{i,r} : i \in [l], r \in \mathcal{R}\}, \mathcal{E} \leftarrow \{E_{i,r} : i \in [l], r \in \mathcal{R}\}$
  return $(gpp^{fb}, gpk^{fb} = (X, Y),\ sfbk = (\alpha, \beta)) \leftarrow$ CDLKeyGen($l$), $\mathcal{E}$, $\Xi$
```

```
CDLVerify*($msg, fbk, \sigma, gpk^{fb}, gpp^{fb}, \mathcal{E}$)
  Let $gpk^{fb} = (X, Y), gpk^{fb}_{i,r} \leftarrow (X, YE_{i,r})$
  return CDLVerify($msg, fbk, \sigma, gpk^{fb}_{i,r}, gpp^{fb}$)
```

```
CDLSign*($msg, fbk, i, r_{b,i}, gsk^{fb}_b, cre_{b,i}, gpk^{fb}_{i,r_{b,i}}, gpp^{fb}$)
  return (CDLSign*($msg, fbk, gsk^{fb}_b, cre_{b,i}, gpk^{fb}_{i,r_{b,i}}, gpp^{fb}$), $i, r$)
```

```
CDLUpdate*
User $b(i, f_b, cre_b = (A_b, B_b, C_b, D_b), gpk^{fb}, gpp^{fb}, \mathcal{E})$        TS$(b, i, r_{b,i}, sfbk, cre_b = (A_b, B_b, C_b, D_b), gpp^{fb}, \Xi)$
                                                                                        $RCT'_i \leftarrow {A_b}^{\xi_{i,r_{b,i}}}; RCT''_i \leftarrow {RCT'_i}^{\alpha}$
                                   $\xleftarrow{RCT'_i, RCT''_i, r_{b,i}}$
  if $\hat{t}_2(RCT'_i, P_5) \neq \hat{t}_2(A_b, E_{i,r_{b,i}})$ then abort
  if $\hat{t}_2(RCT'_i, X) \neq \hat{t}_2(RCT''_i, P_5)$ then abort
  $B_{b,i} \leftarrow B_b RCT'_i; C_{b,i} \leftarrow C_b {RCT''_i}^{f_b}; D_{b,i} \leftarrow {B_{b,i}}^{f_b}$
  return $cre_{b,i} \leftarrow (A_b, B_{b,i}, C_{b,i}, D_{b,i}), r_{b,i}$
```

Figure 1: CDLKeyGen*, CDLSign*, CDLVerify* and CDLUpdate*.

We argue that the security guarantees of CDL [7] carry forward to the modified scheme. There are four modifications to the scheme:

- The inclusion of $\Xi = \{\xi_{i,r} : i \in [l], r \in \mathcal{R}\}$ and $\mathcal{E} = \{E_{i,r} = {P_5}^{\xi_{i,r}} : i \in [l], r \in \mathcal{R}\}$, as part of the private and public parameters.
- Publicly sending RCT'_i and RCT''_i.
- Updating the credential used to sign the signature.
- Updating the public key used to verify the signature.

Based on the original hardness assumptions, for all $i \in [l]$, $r \in \mathcal{R}$, $E_{i,r}$ does not reveal $\xi_{i,r}$. Also RCT'_i and RCT''_i do not reveal $\xi_{i,r}$, or α. None of these steps reveal the secret keys of users f_b, the group manager's secret feedback key $sfbk$, or the set of secret parameters Ξ. Therefore, messages cannot be forged.

Each signature is an instance of the original scheme, with the new public key and credential. Therefore, the untraceability and unlinkability of signatures holds.

4 Our Reputation-Based Messaging scheme

We describe a reputation-based messaging scheme, based on [13]. The following are used in the scheme:

- Algorithms Aggr and Detect, that respectively collate feedback to evaluate a reputation for a user and detect users to be revoked.
- TimeDiscount: $\mathbb{R}_{>0} \rightarrow [0, 1]$, a non-increasing function with TimeDiscount$(0) = 1$.
- A threshold Ψ, that determines the reputation level below which the TS will stop providing secret keys.
- The BBS* and CDL* schemes from III(C)/ III(E).

4.1 Scheme Initialisation

This takes place when the reputation scheme is set up.

1. The TS initialises $\mathsf{TimeDiscount}$, Aggr, Detect and Ψ, BBS* and CDL*. The TS creates a database, to store users' reputation and feedback. Feedback for a message msg will be stored in the list fbl_{msg}.

2. The TS chooses l time intervals $\{\mathbb{T}_i : 0 \leq i \leq l\}$. It picks a set of reputation values, $\mathcal{R} = \{0, 1, ..., r^{max}\}$ with $r^{max} < p$.

3. BKeyGen*(l) outputs gpp^m, gpk^m, $\mathcal{K}$, and smk.

4. CDLKeyGen*$(l, \mathcal{R})$ outputs gpp^{fb}, gpk^{fb}, $\mathcal{E}$, Ξ, and $sfbk$.

4.2 User Registration

This takes place when a user joins the social network.

1. The TS provides the user b with CDL* and BBS*, gpk^m, gpk^{fb}, as well as the public parameters, gpp^m, gpp^{fb}, $\mathcal{K}$ and $\mathcal{E}$.

2. BJoin*(smk, b, gpp^m) is executed by the TS. User b is given their group member secret key for sending messages, $gsk_b^m = (Z_b, x_b)$, over a secure channel.

3. CDLJoin* is executed between the TS with input $(b, gpk^{fb}, sfbk, gpp^{fb})$ and a user b with input (gpk^{fb}, gpp^{fb}). The user b outputs their group member secret key for feedback, $gsk_b^{fb} = f_b$, and credential cre_b. The TS will output cre_b.

4. The TS creates a record in its database for the user b. An entry will be $(b, Z_b, r_{b,i}, F_b, cre_b)$, as well as values of $Z_{b,i}$ for upcoming time intervals.

4.3 Reputation Retrieval

During $\mathbb{T}_i$, a user b retrieves their reputation.

1. To authenticate itself to the TS, b sends a reputation score request signed with BSign*, using $gsk_{b,i}^m$. The TS then uses BOpen*, to discover the signers identity, and finds their current reputation score $r_{b,i}$.

2. If b has been revoked, reputation retrieval will stop. Otherwise the TS computes time discounted reputation scores $(r'_i, r'_{i+1}, ..., r'_{i+d})$ until $r'_{i+d+1} < \Psi$, where $r'_{i+k} = r_{b,i} \cdot \mathsf{TimeDiscount}(t_{i+k} - t_i)$.

3. For all $j = i, ..., (i + d)$, BUpdate* is implemented with input $(b, j, r'_j, smk, gsk_b^m, gpp^m, \mathcal{K})$, to the TS and input $(j, gsk_b^m, gpk^m, gpp^m, \mathcal{K})$, to the user b. The user receives their updated signing key and reputation for time interval $\mathbb{T}_j$, $gsk_{b,j}^m = (Z_{b,j}, x_b)$ and $r_{b,j}$. The TS stores these $Z_{b,j}$ for further use.

4. For all $j = i, ..., (i + d)$, CDLUpdate* is implemented with input $(b, j, r'_j, sfbk, cre_b, gpp^{fb}, \Xi)$, to the TS and input $(j, f_b, cre_b, gpk^{fb}, gpp^{fb}, \mathcal{E})$ to the user b. The user receives their updated credential, and reputation for time interval $\mathbb{T}_j$.

4.4 Public Message Broadcast

A user b with reputation $r_{b,i}$ at time interval $\mathbb{T}_i$ broadcasts a public message M to other users.

1. The user computes $gpk_{i,r_{b,i}}^m = (H_1(K_i^{r_{b,i}})P_1, P_2, U, V, I, W)$, given $gpk^m = (P_1, P_2, U, V, I, W)$,

2. BSign*$(M, i, r_{b,i}, gsk^m_{b,i}, gpk^m_{i,r_{b,i}}, gpp^m)$ is run by user b, to output signature σ on M. The user then broadcasts the message tuple, $msg = (M, \sigma)$.

3. When $msg = (M, \sigma)$ is received, the user determines the current time interval $\mathbb{T}_j$, and checks that $j = i$, and $r < r^{max}$. The user runs BVerify*$(M, \sigma, gpk^m, gpp^m, \mathcal{K})$, and accepts if true.

4.5 Feedback Reporting

A user b_1 gives feedback fbk on a message $msg = (M, \sigma)$ sent by b_2. Let $\mathit{fbr} = (msg, \mathit{fbk})$.

1. b_1 determines the time interval $\mathbb{T}_i$ and computes $gpk^{\mathit{fb}}_{i,r_{b_1,i}}$ from gpk^{fb} (as given in CDLVerify*). b_1 runs CDLSign*$(msg, \mathit{fbk}, i, r_{b_1,i},\ f_{b_1}, cre_{b_1,i}, gpk^{\mathit{fb}}_{i,r_{b_1,i}}, gpp^{fb})$ to output a signature σ_{fb} on fbr. b_1 then sends $(\mathit{fbr}, \sigma_{\mathit{fb}})$ to the TS.

2. The TS identifies the current time interval $\mathbb{T}_j$. They check that $j = i$, and $r < r^{max}$. The TS verifies σ_{fb} with CDLVerify*$(\mathit{fbr}, \sigma_{\mathit{fb}}, gpk^{\mathit{fb}}, gpp^{fb}, \mathcal{E})$, rejecting $(\mathit{fbr}, \sigma_{\mathit{fb}})$ if this outputs false. If the list fbl_{msg} already exists, then for all $(\mathit{fbr}_i, (\sigma_{\mathit{fb}})_i)$ in the list, it checks CDLLink*$((\mathit{fbr}, \sigma_{\mathit{fb}}), (\mathit{fbr}_i, (\sigma_{\mathit{fb}})_i), gpk^{\mathit{fb}}, gpp^{fb})$. If true is ever output, then $(\mathit{fbr}, \sigma_{\mathit{fb}})$ is rejected.

3. If the signature σ is valid, the TS opens msg using BOpen*$(M, \sigma, smk, gpk^m, gpp^m)$ to discover the message's author b_2, to whom it can attribute the feedback. It stores the feedback fbk along with the reputation $r_{b_1,i}$ of b_1. The TS regularly uses Aggr to update its reputation for b_2 in its database.

4. If fbl_{msg} does not exist then this list is created. $(\mathit{fbr}, \sigma_{\mathit{fb}})$ is added to fbl_{msg}.

4.6 User Revocation

Users chosen by Detect are no longer provided with updated secret keys $gsk^m_{b,j}$ and credentials $cre_{b,j}$ during reputation retrieval, meaning they can no longer send public messages, or give feedback.

5 Evaluation

Misbehaving users are not punished by a reputation decrease until the next reputation retrieval, which allows for less frequent communication with the TS. The TimeDiscount function forces users to update regularly, and can be adjusted to take this trade off into account.

Messages could be sent during one time interval and received during the next. So that such messages are accepted, if they are close to the end of the time interval they could be verified with the previous public key.

Constant access to the TS is not required; this is only required for reputation retrieval, and giving feedback.

5.1 Verifiability

The security guarantees of BBS [5] and CDL [7] apply to BBS* and CDL*, as argued in [13] and section 3.5, so that users must be enrolled to send public messages or feedback, for 1(a).

We now argue that all accepted messages and feedback are bound to the correct reputation, for 1(b) and 1(c).

For sending messages, during $\mathbb{T}_i$, if a RCT value from a previous time interval $\mathbb{T}_j$ is used, with reputation r_j, then so the signature is valid they must find r such that $H_1({K_i}^r) = H_1({K_j}^{r_j})$. This is hard in the groups used.

A user may try to increase their reputation by modifying a RCT value. The only way the user can manipulate RCT, while ensuring successful verification, is to raise the RCT to some power a. Then they must find r' such that: $H_1(K_i^r)^a = H_1(K_i^{r'})$. Due to the groups used and the properties of hash functions, this is hard.

When sending feedback, to modify their reputaion using RCT' values for previous or current time intervals, the user must find $(r, A_b^{\xi_{i,r}})$ given $A_b^{\xi_{j,r_j}}$. As all $\xi_{i,r}$ are independent and randomly chosen, this is not possible.

5.2 Correctness

The opening function of BBS [5] ensures feedback can be traced back to its correct subject, for 2(a).

Feedback given more than once per public message can be detected, with CDLLink*, for 2(b).

The scheme can revoke users by no longer allowing them to update their secret keys, so requirement 2(c) is satisfied. Therefore, revocation lists are not necessary.

	Computational Costs for Users excluding Pre–computation	
Sending Public Messages	$9\mathbb{G}_1 + 4\mathbb{G}_3$	$n\mathbb{G}_i$: n group operations in $\mathbb{G}_i$
Verifying Public Messages	$9\mathbb{G}_1 + 2\mathbb{G}_2 + 3\mathbb{G}_3 + 2PO_1$	nPO_i: n pairing operations using the map $\hat{t}_i$
Updating Reputations	$9\mathbb{G}_1 + 4\mathbb{G}_3 + (d+1)(1\mathbb{G}_1 + 2PO_1 + 3\mathbb{G}_4 + 4PO_2)$	$d+1$: number of time intervals secret keys
Signing Feedback	$7 \cdot \mathbb{G}_4$	are updated for during a reputation retrieval.

Table 1: Computational costs of operations performed by users as part of our scheme.

5.3 Privacy

In [13] it is claimed that the security guarantees of BBS [5] imply requirement 3(a), of the privacy of public messages, is satisfied. We claim in section 3.5 that the security guarantees of [7] imply requirement 3(b), of the privacy of feedback, is satisfied.

5.4 Efficiency

5.4.1 Computational cost

In Table I we give the number of operations necessary for processes performed by users regularly. The number of pairings can be reduced by precomputation and collapsing pairings [5].

Of these operations, pairings have the far greatest computational cost [11]. Excluding precomputation, pairing operations are required for verifying messages and updating reputation, but not for signing feedback or sending public messages. Using the Android PBC library [10], for the same security as 128 byte RSA signatures, pairings are undertaken in 27 ms on a Samsung I9100. This would mean pairing operations would take 54ms when verifying a BBS signature.

5.4.2 Communication Overhead

The communication overheads in our scheme when broadcasting public messages and updating reputation include a BBS signature, a reputation value, and the time interval. For the same security as a 128 byte RSA signature, the BBS signature has a length of 192 bytes [5]. The reputation value and time interval have a length of at most 22 bytes and 4 bytes respectively, giving a total of at most 218 bytes.

For feedback the communication overheads include a CDL signature, a reputation value, and the time interval. Using the same groups as for BBS, the CDL signature has a length of 150 bytes, giving a total of 176 bytes.

The average download speed of a smartphone on 4G is 14.7 mBit/s [4]. With this download speed, it will take 0.12ms to download our largest overhead of 218 bytes.

6 Conclusion

To address the lack of trust in PSNs of Strangers, we propose a reputation–based messaging schemeUsers send public messages alongside their reputation, and then give feedback. Users who receive bad feedback consistently are revoked. Public messages are anonymous and unlinkable to other users. If feedback is given once per public message, it is untraceable and unlinkable to all.

To achieve this, we use the group signature scheme BBS* [13] and we modify the DAA scheme [7] in CDL*, to allow users to update their secret key so that it is bound to a reputation score, without a secure channel.

References

[1] Firechat. https://itunes.apple.com/gb/app/firechat/id719829352?mt=8. [Online; accessed 24th-May-2017].

[2] Kik. https://itunes.apple.com/gb/app/kik/id357218860?mt=8. [Online; accessed 13th-March-2017].

[3] Kik already has over 6,000 bots reaching 300 million registered users. https://techcrunch.com/2016/05/11/kik-already-has-over-6000-bots-reaching-300-million-registered-users/. [Online; accessed 1st-August-2017].

[4] Measuring mobile broadband performance in the uk: 4g and 3g network performance. https://www.ofcom.org.uk/__data/assets/pdf_file/0033/79629/ofcom_mbb_performance_report_april_2015.pdf. [Online; accessed 21st-May-2017].

[5] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In *Annual International Cryptology Conference*, pages 41–55. Springer, 2004.

[6] Ernie Brickell, Jan Camenisch, and Liqun Chen. Direct anonymous attestation. In *Proceedings of the 11th ACM conference on Computer and communications security*, pages 132–145. ACM, 2004.

[7] Jan Camenisch, Manu Drijvers, and Anja Lehmann. Universally composable direct anonymous attestation. In *IACR International Workshop on Public Key Cryptography*, pages 234–264. Springer, 2016.

[8] David Chaum and Eugène Van Heyst. Group signatures. In *Workshop on the Theory and Application of of Cryptographic Techniques*, pages 257–265. Springer, 1991.

[9] M. R. Clark, K. Stewart, and K. M. Hopkinson. Dynamic, privacy-preserving decentralized reputation systems. *IEEE Transactions on Mobile Computing*, PP(99):1–1, 2016.

[10] Weiran Liu, Jianwei Liu, Qianhong Wu, and Bo Qin. Android pbc: A pairing based cryptography toolkit for android platform. In *Communications Security Conference (CSC 2014), 2014*, pages 1–6. IET, 2014.

[11] Lukas Malina, Jan Hajny, and Vaclav Zeman. Usability of pairing-based cryptography on smartphones. In *2015 38th International Conference on Telecommunications and Signal Processing (TSP)*, pages 617–621, July 2015.

[12] Félix Gómez Mármol and Gregorio Martínez Pérez. Security threats scenarios in trust and reputation models for distributed systems. *Computers & Security*, 28(7):545–556, 2009.

[13] Siaw-Lynn Ng, Keith Martin, Liqun Chen, and Qin Li. Private reputation retrieval in public - a privacy-aware announcement scheme for vanets. *IET Information Security, DOI: 10.1049/iet-ifs.2014.0316*, 2016.

[14] Zheng Yan, Yu Chen, and Yue Shen. A practical reputation system for pervasive social chatting. *Journal of Computer and System Sciences*, 79(5):556 – 572, 2013.

[15] Ennan Zhai, David Isaac Wolinsky, Ruichuan Chen, Ewa Syta, Chao Teng, and Bryan Ford. Anonrep: towards tracking-resistant anonymous reputation. In *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*, pages 583–596, 2016.