

A fair anonymous submission and review system

Vincent Naessens

Liesje Demuynck

Bart De Decker

Report CW 442, April 2006



Katholieke Universiteit Leuven
Department of Computer Science
Celestijnenlaan 200A – B-3001 Heverlee (Belgium)

A fair anonymous submission and review system

Vincent Naessens

Liesje Demuynck

Bart De Decker

Report CW 442, April 2006

Department of Computer Science, K.U.Leuven

Abstract

Reputation systems play an important role in many Internet communities. They allow individuals to estimate other individual's behavior during interactions. However, a more privacy-friendly reputation system is desirable while maintaining its trustworthiness. This paper presents a fair anonymous submission and review system. The review process is reputation-based and provides better anonymity properties than existing reputation systems. Moreover, the system allows for accountability measures. Anonymous credentials are used as basic blocks.

A Fair Anonymous Submission and Review System

Vincent Naessens¹, Liesje Demuynck ^{*2} and Bart De Decker²

¹ KULeuven Campus Kortrijk, Department of Computer Science,
E. Sabbelaan 53, 8500 Kortrijk, Belgium

² KULeuven, Department of Computer Science,
Celestijnenlaan 200A, 3000 Heverlee, Belgium

Abstract. Reputation systems play an important role in many Internet communities. They allow individuals to estimate other individual's behavior during interactions. However, a more privacy-friendly reputation system is desirable while maintaining its trustworthiness.

This paper presents a fair anonymous submission and review system. The review process is reputation-based and provides better anonymity properties than existing reputation systems. Moreover, the system allows for accountability measures. Anonymous credentials are used as basic blocks.

1 Introduction

In science, peer review is the oldest and best established method of assessing manuscripts, applications for research fellowships and research grants. However, the fairness of peer review, its reliability and whether it achieves its aim to select the best scientist or contributions has often been questioned. It is widely believed that *anonymous reviewing* helps fairness, by liberating reviewers from the fear that openly stated criticism might hurt their careers. A common belief is that some researchers, especially those who are at the start of their careers, may be reluctant to write negative reviews as it could hamper future promotions.

Moreover, Bornmann et al. [1] argue that reviewer's recommendations are frequently biased, i.e. judgements are not solely based on scientific merit, but are also influenced by personal attributes of the author such as author's institutions or names. *Anonymous submissions* can tackle this problem.

On the other hand, Meyer [6] suggests that referees too often hide behind anonymity to turn in *sloppy reviews*; worse, some dismiss contributions unfairly to protect their own competing ideas or products. Even people who are not fundamentally dishonest will produce reviews of unsatisfactory quality out of negligence, laziness or lack of time because they know they can't be challenged. Thus, referees/reviewers must be encouraged to do a decent job. If not, it must still be possible to hold them accountable.

This paper presents a fair anonymous submission and review system. It achieves a reasonable trade-off between the anonymity requirements of the authors and

* Research Assistant of the Research Foundation - Flanders (FWO - Vlaanderen)

reviewers and still allows to identify unfair reviewers. The proposed system aims at improving the fairness of review processes.

The rest of this paper is organized as follows: section 2 describes a general anonymous credential system; these credentials will be used in the submission/review system that is designed in section 3. Section 4 evaluates the system and points to related work. Section 5 concludes with a summary of major achievements.

2 Anonymous credentials

Anonymous credentials allow for anonymous yet accountable transactions between users and organizations. In this section, a simplified version of the Idemix anonymous credential system [2] [5] is presented and extended with a new protocol for credential updating. The Idemix protocols are used as basic building blocks in our system. They typically run on top of an anonymous communication channel.

Nym Registration Protocols. A user is known to an organization under a pseudonym (i.e, *nym*). An individual can establish multiple nym with the same organization. These will all be bound to the user in a specific way such that they cannot be shared with, or borrowed from, other users. Two registration protocols are discussed.

Signed Nym Registration. This protocol is not anonymous. It results in a user U and an organization O obtaining a nym Nym_{UO} , together with the user's signature Sig_{UO} on this nym:

$$U \leftrightarrow O: (Nym_{UO}, Sig_{UO}) = RegSignedNym(Cert_{UA})$$

Note that Sig_{UO} , which can be verified using the external certificate $Cert_{UA}$, provides a provable link between the user's identity and the nym.

Nym Registration. This protocol is used to register a regular nym between a user U and an organization O :

$$U \leftrightarrow O: Nym_{UO} = RegNym()$$

ProofNymPossession Protocol. A user U can also prove to an organization O to be the owner of a nym Nym_{UO} :

$$U \leftrightarrow O: ProofNymPossession(Nym_{UO})$$

Issue Protocol. An organization O can issue a credential $Cred_{UO}$ to a user U . The retrieved credential is known only to the user and cannot be shared. During the issue protocol, the showlimit sl of the credential is set to be either a constant k or unlimited. Also, a number of attributes is incorporated into the

credential. These attributes can be chosen by the organization, or be *jointly created* by both user and organization. In the latter case, the resulting attribute is a random number. It is unknown to O and its value cannot be controlled by either party. $JC(attr)$ defines a jointly created attribute value:

$$U \leftrightarrow I: Cred_{UI} = IssueCred(Nym_{UI}, sl, [attrName = attrValue, \dots])$$

Show Protocol. A user U proves to an organization O that he is in possession of a valid credential $Cred_{UO}$. This action results in a transcript for the organization. During the protocol, several options may be enabled. The user may show his credential with respect to a pseudonym, assuring O that he owns both the credential and the nym. In addition, the resulting transcript may be deanonymizable. Upon fulfillment of a condition $DeanCond$, a trusted deanonymizer is allowed to recover the nym on which the credential was issued. Moreover, the user may disclose some information about the attributes encoded into the credential. He may reveal either an attribute, a property of the attribute, or a verifiable encryption (denoted as $VE_{pk_T}(attr)$) of the attribute. The latter is an encryption of the attribute under the public key of a trusted party T , created in such a way that O can verify its correctness. Finally, the user may decide to sign a message Msg with his credential, providing a provable link between the resulting transcript and the message. Note that different transcripts for the same credential cannot be linked (unless the value of a unique attribute is proved), nor can they be linked to the credential's issue protocol.

$$U \leftrightarrow O: Transcript_{UO} = ShowCred(Cred_{UO}, Nym_{UO}, DeanCond, [AttrProperties], Msg)$$

Update Protocol. A user U can update his credential $Cred_{UO}$ by interacting with its original issuer O . This is particularly useful when the credential has attributes of which the value may change over time. The protocol consists of the user showing his credential to the organization and consecutively receiving a new credential (i.e, the actual update). The new credential is issued on the same nym as the old credential and can be of limited or unlimited show. Its attributes are either the attributes of the old credential or the result of a simple operation f on these attributes (e.g, adding a known value). Apart from the public parameters of the operation f and what is explicitly revealed by the user, the organization does not have any information about the new credential's attributes.

After the execution of the protocol, the old credential will still be valid. Revoking the old credential is often not preferred as this would induce unwanted linkabilities. To overcome this problem, *updatable* credentials are often designed to be one-show. Whenever the credential is updated, its one-show property is renewed. Note that this solution has the drawback that a credential must be updated every time it is shown:

$$U \leftrightarrow O: Transcript_{UO} = ShowCred(Cred_{UO}, Nym_{UO}, DCond, [AttrProps], Msg)$$

$$U \leftrightarrow O: UpdateCred(Cred_{UO}, sl, [AttrChanges])$$

For notational convenience, the update protocol is split up in two separate protocols. We stress that these two operations should always be used together, as they represent a single low-level protocol. More precisely, a *UpdateCred* protocol can never be executed without a preceding *ShowCred* protocol.

Local Deanonimization Protocol. When a credential is shown with its deanonymization option enabled, its resulting transcript $Transcript_{UO'}$ can be used for deanonymization purposes. This protocol is performed by a trusted deanonymizer D , in possession of a secret key sk_D . It results in the pseudonym Nym_{UO} on which the credential was issued and a proof $DeAnProof$ that this nym was retrieved correctly.

$$D: (Nym_{UO}, DeAnProof) = DeanonLocal(Transcript_{UO'})$$

To protect himself from arbitrary deanonymizations, a user specifies a deanonymization condition $DeanCond$ when he shows his credential. Only when this condition is met, D is allowed to perform the local deanonymization.

3 A fair anonymous submission and review system

First, the requirements of the players in the system are described. The requirements analysis results in the design of an anonymous though fair submission and review system that allows for accountability. The roles in the system are described in the second paragraph. Third, we describe the protocols used in the different phases. Finally, complaint handling procedures are discussed.

3.1 Requirements

The conference chair wants to offer a good service to both authors and reviewers in order to attract them. Whereas current submission/review systems mainly focus on the anonymity requirements of the authors, our design considers the concerns of both parties:

- **Anonymous submissions.** Authors must be able to submit papers anonymously. The identity of authors may only be disclosed if one of the following conditions is fulfilled:
 - *The paper is accepted.* The identity of the authors is required to prepare the proceedings.
 - *The paper is already submitted to another conference.* No conference chair accepts double submissions. Authors that submit the same paper to multiple conferences simultaneously are added to a blacklist.
- **Anonymous reviews.** Committee members (i.e, reviewers) must be able to review papers anonymously.

- **Stimulate reviewers.** Committee members must be encouraged to review the papers that are assigned to them. For instance, they can get a discount if they want to attend the conference.
- **Fairness of the review process:**
 - Committee members are not allowed to review the same paper multiple times or to advice on their own paper.
 - The identity of a reviewer must be disclosed if he has written many *unacceptable* reviews.
 - The reviewers’ familiarity with the research domain has an impact on the final outcome of the review process. Reviewers that are very familiar with the subject (i.e. have a reputation in that field) outweigh reviewers that are marginally acquainted with the domain. Therefore, reviewers may not be able to lie about their expertise.

3.2 Roles

Users of the system (U). *Authors* typically submit papers. *Reviewers* are entitled to comment on papers.

Reputation manager (R). The Reputation manager initializes and updates researchers’ reputations. The reputation manager is independent of any conference system.

Conference system infrastructure (C). The Conference system is administered by the Conference Chairman *C*. As depicted in figure 1, the conference system consists of a front end and a back end.

The *front end* of the conference system infrastructure consists of three parts: a submission manager, a review manager and a complaint manager.

The submission manager handles requests from authors. Authors can submit papers and retrieve a contribution token if their paper is accepted. The review manager handles requests from reviewers. Reviewers can register as a committee member. Thereafter, they can review papers. Finally, they can retrieve a token to get a discount when attending the conference. The complaint manager handles complaints from both authors and reviewers.

The *back end* of the conference system consists of a storage manager. The storage manager is responsible for storing submitted papers, reviews and certain types of evidence.

Deanonymization handling infrastructure.

- *Arbiter (A)*. The arbiter’s role is to verify whether a de-anonymization condition is fulfilled.
- *Deanonymizer (D)*. This authority can retrieve the pseudonym under which a credential is issued from a ”show”-transcript.

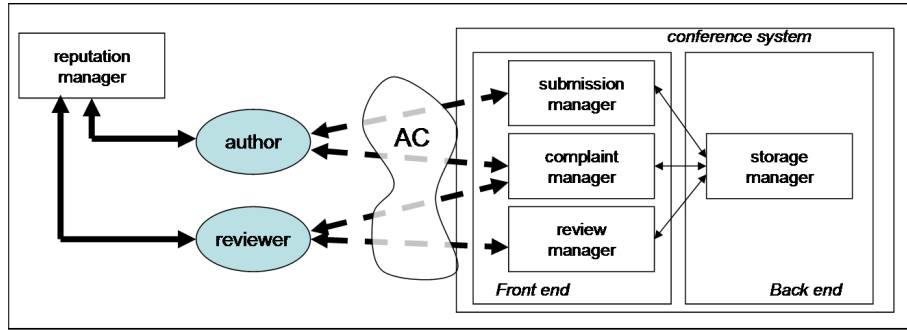


Fig. 1. Overview of the Conference system.

Anonymous communication infrastructure (=AC). The connection between the author/reviewer and the conference system needs to be anonymous.

3.3 Protocols

This section describes the protocols used in different phases. The relation between the protocols are shown in figure 2.

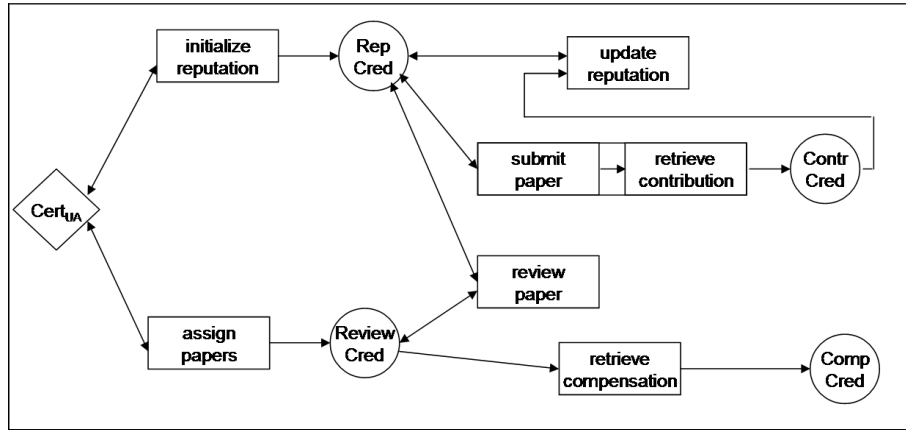


Fig. 2. Overview of actions and credential types.

Initialize_reputation. In this phase, a new researcher (i.e, U) contacts the Reputation Manager R to initialize his reputation in a field. The user first establishes a nym and signs that nym with an external certificate (issued by a

trusted certificate authority A). The Reputation Manager stores the identity proof and issues a reputation credential on the nym. The reputation credential can be shown multiple times to any conference system. Note that an individual can retrieve new reputation credentials as he explores new research domains.

$U \leftrightarrow R : (Nym_{UR}, Sig_{UR}) = RegSignedNym(Cert_{UA})$
 $U \leftrightarrow R : RepCred_{UR} = IssueCred(Nym_{UR}, *, [repField = field, repValue = 0])$
 R : stores $\{Nym_{UR}, Sig_{UR}, Cert_{UA}\}$

Submit_paper(paper). Submitting a paper is conditionally anonymous. If an author submits a *new* paper³, he remains anonymous as long as his paper is not accepted. If the system detects that the paper is already submitted to another conference, his identity will be revealed.

The author first established a nym with the submission manager. During the credential show, the paper is signed, which provably links the paper to the transcript of the credential show. The transcript is deanonymizable. The Submission Manager verifies the credential show and passes it to the Storage Manager. The Storage Manager is responsible for storing the paper, the nym and the transcript.

$U \leftrightarrow C : Nym_{UC} = RegNym()$
 $U \leftrightarrow C : Transcript_{UC} = ShowCred(RepCred_{UR}, Nym_{UC},$
 $\quad 'accepted \vee double_submission', null, \{paper\})$
 C : stores $\{Nym_{UC}, Transcript_{UC}, paper\}$

Retrieve_contribution. After the review process, the Conference Manager publishes a whitelist of nyms. Each nym Nym_{UC} in the list belongs to an author whose paper is accepted. Thereafter, each author can check if his paper is accepted. If so, the author contacts the Submission Manager, proves his identity and to be the owner of a Nym_{UC} in the list of accepted papers. The Submission Manager verifies the proof and issues a contribution token $ContrCred$ to the author. The author can use this credential once (i.e, one-show credential) to update his reputation. The token keeps two attributes: the conference and the research field of the accepted paper.

Note that an author who forgets to check the whitelist can still be traced. A deanonymizer D will eventually reveal the identity of an author. The conference manager must convince the deanonymizer that the paper is really accepted. The strategy to reveal the author behind a submission is discussed in section 3.4.

$U \leftrightarrow C : Sig'_{UA} = ProofIdentity(Cert_{UA})$
 $U \leftrightarrow C : ProofNymPossession(Nym_{UC})$
 $U \leftrightarrow C : ContrCred_{UC} = IssueCred(Nym_{UC}, 1, [contrConf = conf, contrField = field])$

³ A *new* paper is a paper that is not sent previously or simultaneously to another conference.

Update_reputation(field,delta). At this phase, the researcher presents a reputation credential and a contribution credential. Moreover, he proves that the research field in the reputation credential corresponds to the research field in the contribution credential. The reputation credential is updated with a *delta* value. The *delta* value can depend, among others, on the international conference ranking.

$$\begin{aligned}
 U \leftrightarrow R : Nym'_{UR} &= RegNym() \\
 U \leftrightarrow R : Transcript'_{UR} &= ShowCred(RepCred_{UR}, Nym'_{UR}, null, [repField == field], null) \\
 U \leftrightarrow R : Transcript_{UR} &= ShowCred(ContrCred_{UC}, Nym'_{UR}, null, \\
 &\quad [f(contrConf) == delta \wedge contrField == field], null) \\
 U \leftrightarrow R : UpdateCred(RepCred_{UR}, *, [repValue += delta])
 \end{aligned}$$

Assign_papers. In this step, each committee member (i.e, U) contacts the Review Manager to retrieve a review credential. The committee member first establishes a nym and signs that nym with an external certificate. The review credential will be used to control the review process. A review credential contains a set of paper identifiers $revS$. The committee member is expected to review each of the papers that correspond to the identifiers. Moreover, a jointly created review identifier $reviewId$ is assigned to each reviewer.

Note that a *consultation phase*⁴ can precede this step. Committee members can then specify their individual preferences.

$$\begin{aligned}
 U \leftrightarrow C : (Nym'_{UC}, Sig'_{UC}) &= RegSignedNym(Cert_{UA}) \\
 U \leftrightarrow C : ReviewCred_{UC} &= IssueCred(Nym'_{UC}, 1, [revConf = conf, \\
 &\quad revS = S, reviewId = JC(id)]) \\
 C : \text{stores } \{Nym'_{UC}, Sig'_{UC}, Cert_{UA}, S_{rev}\}
 \end{aligned}$$

Review_paper(paperId). The reviewer submits his advice during this phase. An advice typically consists of a list of comments and a score on multiple evaluation criteria (originality, readability...).

The Committee Member first establishes a nym Nym'_{UC} with the Review Manager. He then shows his review credential to prove that the *paper* for which he wants to submit an *advice* was assigned to him. The $reviewId$ is verifiable encrypted with the public key of the deanonymizer. The latter piece of evidence can be used to handle certain types of complaints (see further). The reviewer can also choose to prove that his reputation within the research domain is higher than some predefined levels (10, 20, 30...). This allows the Conference Chairman to measure the familiarity of the reviewer with the research domain of the paper. Note that the transcripts that result from the credential show are provably bound to the nym that is established.

If the advice is submitted successfully, the Review Manager updates the members' review credential (i.e, the paper identifier is removed from the list of as-

⁴ Reviewers could be required to prove that they have enough experience (i.e. have a high reputation) in that field, before they are allowed to bid on a paper.

signed papers). As each review credential is a one-show credential, the old review credential becomes useless. Therefore, a reviewer can not comment multiple times on the same paper.

$$\begin{aligned}
U \leftrightarrow C &: Nym''_{UC} = RegNym() \\
U \leftrightarrow C &: Transcript'_{UC} = ShowCred(ReviewCred_{UC}, null, 'multiple_unacceptable_ \\
&\quad reviews', [paperId \in S \wedge VE_{PK_D}(reviewId) \wedge revConf], null) \\
U \leftrightarrow C &: Transcript''_{UC} = ShowCred(RepCred, Nym''_{UC}, null, \\
&\quad [repField \wedge repValue > x], \{advice, paperId\}) \\
U \leftrightarrow C &: UpdateCred(ReviewCred_{UC}, 1, [revS = revS \setminus paperId]) \\
C &: stores \{Nym''_{UC}, paperId, Transcript'_{UC}, Transcript''_{UC}\}
\end{aligned}$$

Retrieve compensation. After the review deadline, the committee member finalizes his job by contacting the Review Manager. The reviewer first proves to be the owner of a Nym'_{UC} . As explained before, Nym'_{UC} can be provably bound to the identity of a Committee Member. Next, he submits his review credential to prove that the set of remaining papers $revS$ is empty⁵. Hence, the Conference Manager knows which committee members have finalized all reviews. This allows the Conference Manager to send a reminder to committee members that haven't finalized the reviews if the deadline is passed.

Optionally, the Review Manager issues a compensation token. The token can be used to get a discount on the conference fee.

$$\begin{aligned}
U \leftrightarrow C &: ProofNymPossession(Nym'_{UC}) \\
U \leftrightarrow C &: Transcript''_{UC} = ShowCred(ReviewCred_{UC}, Nym'_{UC}, null, \\
&\quad [revS == \emptyset], null) \\
U \leftrightarrow C &: CompCred_{UC} = IssueCred(Nym'_{UC}, 1, null)
\end{aligned}$$

3.4 Complaint handling

Two types of complaints are discussed in this section: complaints related to submissions and complaints related to reviews:

Unacceptable submissions. A submission is unacceptable if it is sent previously/simultaneously to another conference. If so the identity of the author must be revealed⁶. Revealing an author's identity consists of three steps:

- **Decision of Arbiter.** The Complaint Handler sends the suspected paper(s) to the Arbiter. The Arbiter first verifies the validity of the papers w.r.t. the transcripts. He then verifies whether the papers are really very similar. The Arbiter returns his signed decision. If the paper is unacceptable, the Complaint Handler informs the Deanonymizer.

⁵ Note that a committee member can only retrieve a compensation once as the review credential is a one-show credential.

⁶ Note that this strategy can also be used to reveal the author behind an accepted paper that forget to check the whitelist.

- **Disclosing Nym.** The Deanonymizer receives a signed message from the Complaint Handler. The message contains the Arbiter’s decision (i.e. ”Unacceptable”), the paper and the transcript. The Deanonymizer verifies the advice, and if positive, locally deanonymizes the transcript. He then returns the nym Nym_{UR} and a deanonymization transcript to the Complaint Handler. The Deanonymizer also stores the Arbiter’s signed decision.
- **Revealing identity.** The Complaint Handler forwards the evidence to the Reputation Manager and orders the Reputation Manager to reveal the identity of the user behind the Nym_{UR} . The Complaint Handler stores the evidence that proves the link between the author and the submissions.

Unacceptable reviews. An unacceptable review policy can be worked out by the Conference Manager. Note that both a conference chairman as well as an author (when receiving feedback) can initiate a complaint of this type. The complaint handling procedure consists of three steps:

- **Decision of Arbiter.** (see above⁷)
- **Disclosing review identifier.** If the review is unacceptable, the Complaint Handler convinces the deanonymizer to reveal the review identifier $reviewId$ that is verifiably encrypted in $Transcript'_{UC}$. The deanonymizer then returns the review identifier to the Complaint Handler.
- **Revealing identity (optionally)** If multiple unacceptable reviews correspond to the same review identifier, the Complaint Handler sends the evidence and $Transcript'_{UC}$ to the deanonymizer. The deanonymizer deanonymizes $Transcript'_{UC}$ and returns Nym'_{UC} and the deanonymization transcript to the Complaint Manager. The Conference system keeps a provable mapping between Nym'_{UC} and the identity of the reviewer.

4 Evaluation

This section focuses on the anonymity/trust properties in the system. The conference management system creates a trusted environment for authors, reviewers and conference managers.

Authors. An author may trust that his submission will not be linked to his identity (even not by the conference chairman) as long as his paper is not accepted and given that he did not submit his contribution to another conference. Four entities are required to reveal the identity of an author, namely C , A , D and R . D will only deanonymize the transcript after permission of an arbiter. However, trust can even easily be distributed between multiple deanonymizers D_i and arbiters A_j . This implies that a set of arbiters decide whether the deanonymization condition is fulfilled and a set of deanonymizers is required to reveal the nym_{UR} behind the transcript.

⁷ Note that in this case, the suspected review is sent to the Arbiter.

Reviewers. Except in very unusual circumstances, the identity of the reviewers involved in the review of any given paper is not known by any party. The identity of reviewers will only be revealed if they wrote several reviews of inferior quality. *C*, *A* and *D* are required to disclose the identity of a reviewer. Again, trust can be distributed between multiple arbiters and deanonymizers.

Conference chairman. The conference chairman provides a fair review process. Although the conference manager does not know the identity of the reviewer of a paper, a referee can not lie about his expertise in research domains. This improves the fairness of review processes.

Reputation manager. Researchers can only update their reputation if they retrieved a contribution credential. Consequently, the reputation manager needs to rely on conference managers. However, the reputation manager will only increase the users reputation value slightly if the contribution credential was issued by a low ranked conference.

5 Discussion

Anonymous reputation systems. Reputation systems [7] [3] [4] already play an important role in Internet communities like eBay. Unfortunately, the design of current reputation systems allow to generate user profiles. Ultimately, the user can be uniquely identified. The main problem is that the reputation is tightly-coupled to a pseudonym in many systems. Our design does not bind a reputation value to a single pseudonym. Thus, multiple proofs of the same reputation cannot be linked.

Moreover, our system enables to prove properties of the reputation value (for instance, $value > 10$). This implies that a user with a very high reputation value can still convince a conference chairman without being uniquely identified.

We have demonstrated the use of updatable credentials within an anonymous reputation system. It is clear that this new concept is useful in many applications. In particular in applications where the value of a credential's attribute depends on external factors and hence may change over time. In its low-level implementation, a show protocol precedes the actual update protocol. In this regard, its computational cost is slightly more than the cost of an individual show or issue protocol, but significantly less than the cost of both primitives together.

Reputation credential versus review credential. Both credentials store attributes whose value can change. However, both types have a slightly different implementation. Whereas reputation credentials are multi-show credentials, review credentials are one-show. Both strategies have advantages that are exploited in the conference system.

An unlimited-show credential allows users to prove (properties of) attributes unlimitedly. Hence, researchers can prove properties of their expertise without

having their credential to be updated.

One-show credentials prevent subjects to use the credential multiple times. Thus, a committee member cannot present an older version of the review credential multiple times (i.e., $revCred_{UC}$). This prevents him to submit more than one review for the same paper. However, a reviewer can use an old reputation credential (i.e., $repCred_{UR}$) when reviewing a paper. Nevertheless, as newer reputation credentials have a higher value, he will not be inclined present an older one.

Handling multiple research domains. Researchers can have expertise in multiple research domains. Similarly, a paper can present experiences in multiple research domains. Hence, a reviewer must be able to prove his familiarity with each of these domains.

In a straightforward solution, the researcher retrieves a reputation credential from the Reputation Manager for each domain in which he is involved and uses a subset of these credentials at each review. This has many disadvantages. First, a mature researcher may have to store many credentials. Second, a lot of overhead is introduced when multiple reputation credentials have to be shown.

Another solution foresees multiple domains and values in one credential. However, as many research domains exist, the credential size will also be large.

A hybrid solution defines a set of general research domains. Each domain is split in subdomains. A credential can be retrieved for each domain. One credential stores a researchers' reputation value within each subdomain. For instance, the ACM Computing Classification System can be used to fix sub(domains).

Misleading reputations. If an individual has not made any relevant contributions within the last years, his reputation value may be misleading. This can compromise the fairness of the review process. To tackle this problem, the reputation credential could also keep the dates and contribution values of the most recent publications. These attributes can also be used to calculate the user's final reputation value. Hence, reputation credentials that are not updated recently decrease implicitly:

$$f(\text{value}, [\text{year}_1, \text{value}_1], [\text{year}_2, \text{value}_2], [\text{year}_3, \text{value}_3]) > x$$

Conflicting interests. Although a conference manager can demand from committee members to indicate conflicting interests during the consultation phase, a committee member can still neglect this demand. Hence, a committee member could be assigned his own paper. However, the authors behind accepted papers are identified. Moreover, the Conference Chairman knows the set of reviewers that did comment on a paper. Consequently, the Conference Chairman can check after the review process whether a conflict of interests occurred. If so, the conference chairman can decide to revise the acceptability status of the paper.

6 Conclusions

This paper presented a fair anonymous submission and review system. The system provides a trusted environment for authors, reviewers and conference chairmen. The review process is reputation-based and allows for accountability measures. We also demonstrated the use of updatable credentials within an anonymous reputation system. It is clear that this new concept can be extended to many other application domains where the value of a credential's attribute depends on external factors and hence may change over time.

References

1. L. Bornmann, H. D. Daniel, Reliability, fairness and predictive validity of committee peer review. Evaluation of the selection of post-graduate fellowship holders by the Boehringer Ingelheim Fonds.B.I.F. In *FUTURA 19*, p. 7-19, 2004.
2. Jan Camenisch, Els Van Herreweghen. Design and Implementation of the Idemix Anonymous Credential System. Research Report RZ 3419, IBM Research Division, June 2002. Also appeared in *ACM Computer and Communication Security*, 2002.
3. C. Dellarocas. Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior. In *Proceedings of the ACM Conference on Electronic Commerce*, 2000, 150-157.
4. R. Dingledine, N. Mathewson, and P. Syverson. Reputation in P2P Anonymity Systems. In *Proceedings of Workshop on Economics of Peer-to-Peer Systems*, June 2003.
5. E. Van Herreweghen. Unidentifiability and Accountability in Electronic Transactions. PhD Thesis, KULeuven, October 2004.
6. B. Meyer. Open refereeing: Why I sign my reviews. <http://se.ethz.ch/meyer/publications/>
7. S. Steinberger. Privacy-enhancing Reputation Systems for Internet Communities. In *Proceedings of the 21th IFIP International Conference on Information Processing: Security and privacy in dynamic environments*, to appear, May 2006.