A Fair Anonymous Submission and Review System

Vincent Naessens[1], Liesje Demuynck[2,⋆], and Bart De Decker[2]

[1] KULeuven Campus Kortrijk, Department of Computer Science,
E. Sabbelaan 53, 8500 Kortrijk, Belgium
vincent.naessens@kuleuven-kortrijk.be
[2] KULeuven, Department of Computer Science,
Celestijnenlaan 200A, 3000 Heverlee, Belgium
{liesje.demuynck, bart.dedecker}@cs.kuleuven.be

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. Some researchers 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 institution or name. *Anonymous submissions* can tackle this problem.

On the other hand, Meyer [5] 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.

[⋆] Research Assistant of the Research Foundation - Flanders (FWO - Vlaanderen).

H. Leitold and E. Markatos (Eds.): CMS 2006, LNCS 4237, pp. 43–53, 2006.

This paper presents a fair anonymous submission and review system. It achieves a reasonable trade-off between the anonymity requirements of the authors and 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, 7] is presented and extended with a new protocol for credential updating. The protocols are used as basic building blocks in our system. They typically run over an anonymous communication channel.

RegNym Protocols. An individual can establish multiple non-transferable pseudonyms (i.e. *nyms*) with the same organization. Two registration protocols are discussed:

- *$U \leftrightarrow O$: (Nym_{UO}, Sig_{UO}) = RegSignedNym($Cert_{UA}$).* During the signed nym registration protocol, the user signs the established Nym_{UO} with his signature key, which is certified through an external certificate (which links the user's public key with his identity). Hence, the organization holds a provable link between the nym and the identity certified by the certificate.
- *$U \leftrightarrow O$: Nym_{UO} = RegNym().* The (ordinary) Nym Registration protocol is used to register a regular nym between a user U and an organization O.

ProofNymPossession Protocol. $U \leftrightarrow O : ProofNymPossession(Nym_{UO})$. A user U can prove to an organization O to be the owner of a nym Nym_{UO}.

Issue Protocol. *$U \leftrightarrow I$: $Cred_{UI}$ = IssueCred(Nym_{UI}, sl, $\{$attrName = attrValue, ...$\}$).* An issuer I can issue a credential $Cred_{UI}$ to a nym Nym_{UI}. 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 embedded into the credential.

Show Protocol. *$U \leftrightarrow V$: $Transcript_{UV}$ = ShowCred($Cred_{UI}$,[Nym_{UV}], [DeanCond], [AttrProperties], [Msg]).* A user U proves to a verifier V that he is in possession of a valid credential Cred$_{UI}$. This action results in a transcript for the verifier. During the protocol, several options may be enabled. The user may show his credential with respect to a pseudonym Nym_{UV}, by which he is known to V. This provably links the transcript 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 or a property of the attribute, and may decide to sign a message *Msg* with his credential; creating a provable link between the 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.

Update Protocol. A user U can update his credential $Cred_{UI}$ by interacting with its original issuer I. 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 I and consecutively receiving a new credential (i.e. the actual update). The new credential is issued on the same nym as the old credential. 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 issuer does not have any information about the new credential's attributes. Note that the old credential will still be valid after the execution of the protocol unless it is a one-show credential. Note also that an *UpdateCred* protocol can never be executed without a preceding *ShowCred* protocol.
$U \leftrightarrow I$: $Transcript_{UI} = ShowCred(Cred_{UI}, Nym_{UI}, [DCond], [AttrProps], [Msg])$
$U \leftrightarrow I$: $UpdateCred(Cred_{UI}, sl, [AttrChanges])$

Local Deanonymization Protocol. D: $(Nym_{UI}, DeAnProof) = DeanonLocal(Transcript_{UV})$. If a credential show is deanonymizable, the pseudonym Nym_{UI} on which the credential was issued can be revealed by a trusted deanonymizer D. *DeAnProof* proofs the link between the transcript and the nym. D is only allowed to perform the deanonymization when *DeanCond* is met.

3 A Fair Anonymous Submission and Review System

First, the requirements and roles are described. Next, we describe the protocols used in the different phases. Finally, complaint handling procedures are discussed.

3.1 Requirements and Roles

Requirements. Whereas current conference systems mainly focus on the anonymity requirements of the authors, our design considers the concerns of all users:

- *Anonymity requirements.* Committee members (i.e. reviewers) must be able to review papers anonymously. Similarly, authors must be able to submit papers anonymously. The identity of authors may only be disclosed when the paper is accepted (i.e. the identity of the authors is required for preparing the program) or when the paper is submitted simultaneously to another conference (i.e. no conference chair accepts double submissions).
- *Requirements related to fairness.* First, committee members are not allowed to review the same paper multiple times or to advice on their own papers.

Second, the identity of a reviewer can be disclosed if he has written many *unacceptable* reviews. Third, the reviewers' familiarity with the research domain must have an impact on the final outcome of the review process. Therefore, reviewers may not be able to lie about their expertise. Finally, committee members must be encouraged to review the papers that are assigned to them. For instance, they can get a discount on the conference fee.

Roles. *Users (U)* are either authors or reviewers. The *Reputation manager (R)* initializes and updates their reputations. The reputation manager is independent of any conference system.

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 system 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 when 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 discount token. 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.

There is also a *deanonymization infrastructure*. It consists of an *Arbiter (A)* and a *Deanonymizer (D)*. A's role is to verify whether a de-anonymization condition is fulfilled. D can retrieve the pseudonym under which a credential is issued from a "show"-transcript. An *anonymous communication infrastructure (=AC)* is required as the connection between U and the conference system needs to be anonymous.

3.2 Protocols

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

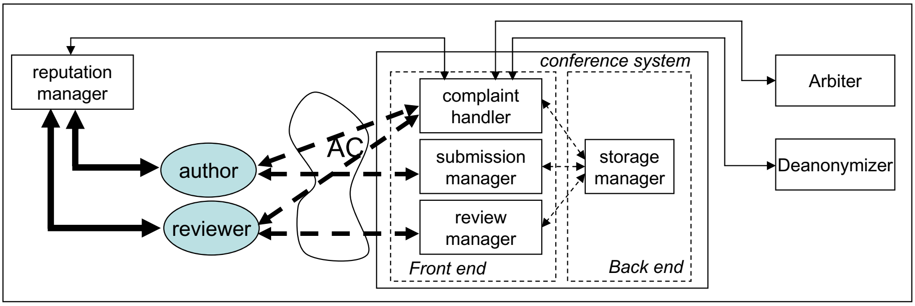

Fig. 1. Overview of the Conference Management System

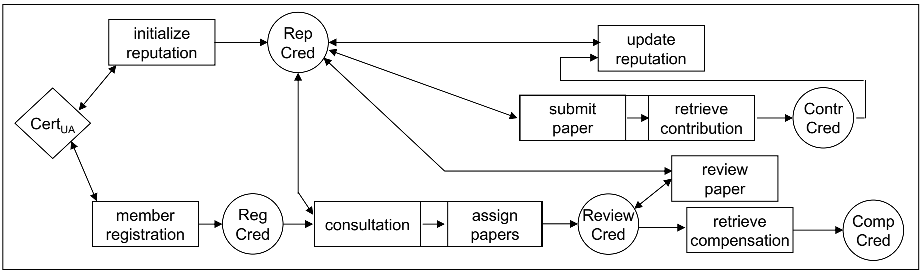

Fig. 2. Overview of actions and credential types

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). R stores the identity proof and issues a reputation credential on the nym. It can be shown unlimitedly. 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. When an author submits a *new* paper[1], he remains anonymous as long as his paper is not accepted. The author first establishes 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 stores the paper, the nym and the transcript.

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

Retrieve_contribution. After the review process, the Conference Manager publishes a whitelist containing the titles of accepted papers. Each title contains a link to a nym Nym_{UC} used during submission. The author can check whether his paper is accepted. If so, he contacts the Submission Manager, proves his identity and also proves to be the owner of the corresponding Nym_{UC}. The Submission Manager verifies the proof and issues a contribution token $ContrCred$ to the

[1] A *new* paper is a paper that is not sent previously to another conference. When the paper has already been submitted to another conference, and this is detected, the author's identity will be revealed.

author. The author can use this one-show credential once, to update his reputation. It contains two attributes: the conference id and the research field of the accepted paper.

Note that an author who forgets to check the whitelist can still be traced. A deanonymizer will eventually reveal the identity of an author. Hereto, 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 3.3.

$U \leftrightarrow C : \mathit{Sig}'_{UR} = \mathit{ProofIdentity}(\mathit{Cert}_{UA})$
$U \leftrightarrow C : \mathit{ProofNymPossession}(\mathit{Nym}_{UC})$
$U \leftrightarrow C : \mathit{ContrCred}_{UC} = \mathit{IssueCred}(\mathit{Nym}_{UC},\ 1,$
$\quad \{\mathit{contrConf} = \mathit{conf},\ \mathit{contrField} = \mathit{field}\})$

Update_reputation(field,delta). 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 an international ranking.

$U \leftrightarrow R : \mathit{Transcript}_{UR} = \mathit{ShowCred}(\mathit{RepCred}_{UR},\ \mathit{null},\ \mathit{null},$
$\quad \{\mathit{repField} == \mathit{field}\},\ \mathit{null})$
$U \leftrightarrow R : \mathit{Transcript}'_{UR} = \mathit{ShowCred}(\mathit{ContrCred}_{UC},\ \mathit{null},\ \mathit{null},$
$\quad \{f(\mathit{contrConf}) == \mathit{delta} \wedge \mathit{contrField} == \mathit{field}\},\ \mathit{null})$
$U \leftrightarrow R : \mathit{UpdateCred}(\mathit{RepCred}_{UR},\ *,\ [\mathit{repValue}{+}{=}\mathit{delta}])$

Member_registration. In this step, each committee member (i.e. U) contacts the Review Manager to retrieve a registration credential. The committee member first establishes a Nym'_{UC} and signs that nym with an external certificate. The registration credential will be used to control the consultation process.

$U \leftrightarrow C : (\mathit{Nym}'_{UC},\ \mathit{Sig}'_{UC}) = \mathit{RegSignedNym}(\mathit{Cert}_{UA})$
$U \leftrightarrow C : \mathit{RegCred}_{UC} = \mathit{IssueCred}(\mathit{Nym}'_{UC}, 1,\ \{\mathit{revConf} = \mathit{conf}\})$
C : stores $\{\mathit{Nym}'_{UC}, \mathit{Sig}'_{UC}, \mathit{Cert}_{UA}\}$

Consultation(preferences). After the registration deadline, registered members can specify their individual *preferences* anonymously (i.e. relative to a Nym''_{UC}). Reviewers are required to prove that they have enough experience (i.e. a high reputation) in that field, before they are allowed to bid on a paper. The Review Manager processes the preferences of all committee members.

$U \leftrightarrow C : \mathit{Nym}''_{UC} = \mathit{RegNym}()$
$U \leftrightarrow C : \mathit{Transcript}^{1}_{UC} = \mathit{ShowCred}(\mathit{RegCred}_{UC},\ \mathit{Nym}''_{UC},\ \text{'}\mathit{multiple\_}$
$\quad \mathit{unacceptable\_reviews}\text{'},\ \{\mathit{revConf} == \mathit{conf}\},\ \mathit{preferences})$
$U \leftrightarrow C : \mathit{Transcript}^{2}_{UC} = \mathit{ShowCred}(\mathit{RepCred}_{UR},\ \mathit{Nym}''_{UC},\ \mathit{null},$
$\quad \{\mathit{repField} == \mathit{field} \wedge \mathit{repValue} > x\}, \mathit{null})$
C : stores $\{\mathit{Nym}''_{UC},\ \mathit{Transcript}^{1}_{UC},\ \mathit{preferences}\}$

Assign_papers. In this step, each committee member (i.e. U) contacts the Review Manager to retrieve a review credential. The committee member proves to be the owner of a Nym''_{UC} (that was established in the consultation phase). 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. It is clear that *revS* depends on the preferences of the reviewer.

$U \leftrightarrow C$: $ProofNymPossession(Nym''_{UC})$
$U \leftrightarrow C$: $ReviewCred_{UC} = IssueCred(Nym''_{UC}, 1, \{revConf = conf, revS = S\})$

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

The Committee Member shows his review credential to prove that the *paper* for which he wants to submit an *advice* was assigned to him. The reviewer can also choose to prove that his reputation is higher than some predefined level. This allows the Conference Chairman to measure the familiarity of the reviewer with the research domain of the paper. If the advice is submitted successfully, the Review Manager updates the members' review credential (i.e. the *paperId* is removed from the list of assigned papers). As each review credential is a one-show credential, the old review credential becomes useless. Therefore, a reviewer cannot comment multiple times on the same paper.

$U \leftrightarrow C$: $Transcript^3_{UC} = ShowCred(ReviewCred_{UC}, null,$
$\quad$ $'unacceptable\_review', \{paperId \in revS \wedge revConf == conf\}, null)$
$U \leftrightarrow C$: $Transcript^4_{UC} = ShowCred(RepCred_{UR}, null, null,$
$\quad$ $\{repField == field \wedge repValue > x\}, \{advice, paperId\})$
$U \leftrightarrow C$: $UpdateCred(ReviewCred_{UC}, 1, \{revS = revS \setminus paperId\})$
C : stores $\{advice, paperId, Transcript^3_{UC}\}$

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 empty. Hence, the Conference Chair knows which committee members have finalized all reviews. This allows the Conference Manager to send a reminder to members that haven't finalized the reviews if the deadline is passed. Optionally, the Review Manager issues a compensation token that can be used to get a discount on the conference fee.

$U \leftrightarrow C$: $ProofNymPossession(Nym'_{UC})$
$U \leftrightarrow C$: $Transcript^5_{UC} = ShowCred(ReviewCred_{UC}, Nym'_{UC},$
$\quad$ $null, \{revS == \emptyset\}, null)$
$U \leftrightarrow C$: $CompCred_{UC} = IssueCred(Nym'_{UC}, 1, null)$

3.3 Complaint Handling

Two types of complaints are discussed in this section: complaints related to submissions and complaints related to reviews. A *submission is unacceptable* if it is sent previously/simultaniously to another conference. If so the identity of the author must be revealed[2]. It consists of three steps:

- **Decision of Arbiter (A).** The Complaint Handler sends the suspected paper(s) to A. A verifies whether the papers are really very similar and returns his signed decision. If so, the Complaint Handler informs D.
- **Disclosing Nym.** D receives a signed message from the Complaint Handler. The message contains A's decision, the paper and the $Transcript_{UC}$. D verifies the decision, and if positive, deanonymizes the transcript. He then returns Nym_{UR} and a deanonymization transcript to the Complaint Handler.
- **Revealing identity.** The Complaint Handler forwards the evidence to the Reputation Manager R and orders R 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.

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. It consists of three steps:

- **Decision of Arbiter.** (see above[3])
- **Disclosing review identifier.** If the review is unacceptable, the Complaint Handler convinces D to deanonymize $Transcript^3_{UR}$. D then returns the Nym''_{UC} and the deanonymization transcript.
- **Revealing identity (optionally)** If multiple unacceptable reviews correspond to the same Nym''_{UC}, the Complaint Handler sends the evidence and $Transcript'_{UC}$ to D. D deanonymizes $Transcript^1_{UC}$ 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 of the system. The conference management system creates a trusted environment for all players.

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 not double submitted. 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 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.

[2] Note that this strategy can also be used to reveal the author of an accepted paper who forgot to check the whitelist.

[3] Note that in this case, the suspected review is sent to the Arbiter.

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.

Although the *conference manager* does not know the identity of the reviewer of a paper, a referee can not lie about his expertise in a research domain. This improves the fairness of the review proces.

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 [3, 4, 6] already play an important role in Internet communities like eBay. Unfortunately, the design of current reputation systems allows 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, $\textit{value} > \textit{10}$). 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. 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.

The *reputation credential* and the *review credential* contain 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. $\textit{ReviewCred}_{UC}$). This prevents him to submit more than one review for the same paper. However, a reviewer can use an old reputation credential (i.e. $\textit{RepCred}_{UR}$)

when reviewing a paper. Nevertheless, as newer reputation credentials have a higher value, he will not be inclined present an older one.

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).

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: $f(value, [year_1, value_1], [year_2, value_2], [year_3,value_3]) > x$.

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 stores the nyms Nym''_{UC} of reviewers that did comment on a paper. He also stores the corresponding $Transcript^1_{UC}$ which can be deanonymized by D (and which can lead to the identity of the reviewer). Consequently, D can check after the review process whether a conflict of interests occurred. If so, he informs the Conference Chairman who, on his turn, can decide to revise the acceptability status of the paper. Alternatively, authors can be demanded to indicate conflicts of interests. However, the latter strategy may reduce the anonymity set of authors.

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 *Proc. of Workshop on Economics of Peer-to-Peer Systems*, June 2003.
5. B. Meyer. Open refereeing. `http://se.ethz.ch/~meyer/publications/`
6. S. Steinbrecher. 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.
7. E. Van Herreweghen. Unidentifiability and Accountability in Electronic Transactions. PhD Thesis, KULeuven, October 2004.