Privacy-friendly Incentives and their Application to Wikipedia

(Extended Version)

Jan Camenisch* Thomas Groß* Peter Hladky† Christian Hoertnagl*

May 5, 2010

Abstract

Double-blind peer review is a powerful method to achieve high quality and thus trustworthiness of user-contributed content. Facilitating such reviews requires incentives as well as privacy protection for the reviewers. In this paper, we present the concept of privacy-friendly incentives and discuss the properties required from it. We then propose a concrete cryptographic realization based on ideas from anonymous e-cash and credential systems. Finally, we report on our software's integration into the MediaWiki software.

1 Introduction

We, as users, all rely increasingly on information on the Internet, ranging from stock quotes and financial news to medical information. Also, businesses and organizations (including governments) rely on information on the Internet to make their decisions—including, for instance, court cases and financial investments. It is therefore crucial that this information can be trusted to be correct.

Information provided by organizations is typically considered trustworthy because organizations are trusted to have quality assurance processes in place. Moreover, they can be held liable for publishing incorrect information. An increasing part of the Internet's content is *user-contributed*. Here, assessing the trustworthiness of the information is much more difficult, because the contributing users are typically barely known and can easily be impersonated. Also, as they can hardly be held liable, users sometimes contribute wrong information, on purpose. Examples range from discrediting other users to manipulating votes or markets.

Sites such as Wikipedia try to address this problem by establishing a user's reputation. This is normally done by registration and *identification of the users*, sacrificing users' privacy for the quality of their contributions. For instance, Citizendium, a new electronic encyclopedia project, only accepts contributors who are registered with full curriculum vitae and proof of identity. The contributors must consent to obligatory disclosure of their Personal Identifiable Information (PII). However, users often prefer to be anonymous or pseudonymous, when contributing contents or commenting on other contributions. In fact, it is crucial for protecting all our on-line privacy to be able to interact with such wide on-line communities in an anonymous or pseudonymous way. Moreover pseudonymous interactions generally seem to guarantee higher quality of contributions.

An additional mechanism for quality assurance is *distributed moderation or rating* as, for instance, used by in Slashdot.org or Apple's App store for the iPhone. Distributed moderation is typically done by rating, tagging, and reviewing of contributions or, in other words, by adding meta-data of the user community itself. It seems that such systems can quickly and consistently separate high and low quality comments in an online conversation [21], on the one hand, but also that the quantity and quality of meta-data may not be sufficient in practice unless the users are given sufficient incentives, on the other hand. The latter was also observed in an experiment made on the IBM Intranet as part of the PrimeLife project [1]. Incentives could be in the form of

*IBM Research, Zurich Rüschlikon, Switzerland

†Swiss Federal Institute of Technology (ETH), Zurich, Switzerland

monetary payments (e.g., micro-payments or points that can be redeemed later for a book or CD), valuations such as gaining reputation (cf. eBay), or in the form of side-effects (e.g., as games with a purpose [26]).

In conclusion, we need an on-line collaboration system that, on the one hand, protects the privacy of the users and, on the other hand, enhances the quality by giving incentives for reviews and moderation. For the latter, we need of course to ensure that the privacy offered cannot be abused. For instance, it must not be possible that one and the same person provides the original contribution and then also does all the moderation and reviews. In the paper, we first investigate the requirements and then provide a system that, on the one hand, offers maximal privacy to the users and, on the other hand, allows for providing incentives and the establishment of reputations. Our system draws on unlinkable pseudonyms, anonymous credentials and e-cash.

Contributions. We specify the first privacy-friendly incentive system with strong privacy protection and accountability. The system not only covers incentives and reputation, but also separation of duty, role-based and attribute-based entitlement policies. We provide a cryptographic realization based on abstract interfaces with zero-knowledge proofs of knowledge, anonymous credential systems and anonymous e-cash as primitives. Our system can be instantiated in an SRSA as well as in an ECC setting.

We have implemented our incentive system for Wikipedia based on the Identity Mixer cryptographic library (SRSA). It can be used with any other on-line collaboration platform. We intend to make our source code publicly available.

2 Preliminaries

In this section we describe the abstract interfaces of the cryptographic primitives we employ, mostly following Bangerter et al. [4]. Our actual implementation uses and extends the Identity Mixer cryptographic library [20] which offers these primitives.

2.1 Commitment Schemes

A commitment scheme allows one to commit to a message m from some domain (typically $\mathbb{Z}q$ for some prime q). The interface is as follows.

- $C \leftarrow \mathsf{Commit}(m, r)$
- 0 or 1 $\leftarrow \mathsf{VerifyCommit}(C, m, r)$

The interface can be instantiated by the Pedersen commitment scheme [23] or the Integer commitment scheme by Damgård and Fujisaki [15]. For the Pedersen scheme, public parameters are a group G of prime order q, and generators $(g_0, \ldots, g_l)$. In order to commit to the values $(m_1, \ldots, m_l) \in \mathbb{Z}q^l$, pick a random $r \in \mathbb{Z}q$ and set

$$C \leftarrow \mathsf{Commit}((m_1, \ldots, m_l), r) = g_0^r \prod_{i=1}^{l} g_i^{m_i}.$$

2.2 Zero-Knowledge Proofs and Σ-Protocols

When referring to the zero-knowledge proofs of knowledge of discrete logarithms and statements about them, we will follow the notation introduced by Camenisch and Stadler [14] and formally defined by Camenisch, Kiayias, and Yung [11].

For instance, $PK\{(a, b, c) : y = g^a h^b \wedge \tilde{y} = \tilde{g}^a \tilde{h}^c\}$ denotes a "*zero-knowledge Proof of Knowledge of integers a, b, c such that $y = g^a h^b$ and $\tilde{y} = \tilde{g}^a \tilde{h}^c$ holds*," where $y, g, h, \tilde{y}, \tilde{g}$, and $\tilde{h}$ are elements of some groups $G = \langle g \rangle = \langle h \rangle$ and $\tilde{G} = \langle \tilde{g} \rangle = \langle \tilde{h} \rangle$. Following the approach of Bangerter et al., the PK notation accepts abstract predicates as inputs. For instance, $PK\{(m, r) : \mathsf{VerifyCommit}(C, m, r)\}$ denotes the proof

of representation of a commitment. SPK denotes a signature proof of knowledge, that is a non-interactive transformation of a proof with the Fiat-Shamir Heuristic [17].

2.3 Signature Scheme for Anonymous Credentials

Bangerter et al. [4] formalize anonymous credential systems as an abstract signature interface. The signer is an Issuer I with key pair (sk_{I}, pk_{I}).

- $(sk_{\mathsf{I}}, pk_{\mathsf{I}}) \leftarrow \mathsf{SetupSign}(\ell)$: Key generation for the Issuer I.
- $(\sigma)() \leftarrow \mathsf{HiddenSign}((C_1, \ldots, C_{l'}), (m_{l'+1}, \ldots m_l), r; pk_{\mathsf{I}})(sk_{\mathsf{I}})$:
  Issuer I signs hidden messages $(m_1, \ldots, m_{l'})$ in commitments $(C_1, \ldots, C_{l'})$ as well as known messages $(m_{l'+1}, \ldots, m_l)$. The user completes the signature σ with the commitment randomness r.
- 0 or $1 \leftarrow \mathsf{VerifySign}(\sigma, (m_1, \ldots, m_l); pk_{\mathsf{I}})$:
  Predicate to verify a signature σ by Issuer I on messages $(m_1, \ldots, m_l)$.
- 0 or $1 \leftarrow \mathsf{VerifySignPred}(\sigma, (m_1, \ldots, m_l), \mathsf{AttrPredicate}; pk_{\mathsf{I}})$:
  Verifies additionally that the efficiently provable predicate $\mathsf{AttrPredicate}$ over the messages $(m_1, \ldots, m_l)$ is fulfilled.

This abstraction contains two key points: First, it provides a $\mathsf{HiddenSign}()$ function that allows an Issuer I to sign committed values $C_i = \mathsf{Commit}(m_i, r), 1, \ldots, i, \ldots l'$ without knowledge of the hidden values m_i. Second, the framework provides a predicate $\mathsf{VerifySign}()$ that allows for a verification of signatures in zero-knowledge proofs of knowledge. Thirdly, we offer an additional predicate $\mathsf{VerifySignPred}()$ to verify attribute statements over the attributes of signature σ in zero-knowledge proofs.

2.4 E-Coin Schemes

Our construction uses simple e-coins as basic building block. We reference compact e-cash [10] for a formal set of definitions.

- $(sk_{\mathsf{B}}, pk_{\mathsf{B}}) \leftarrow \mathsf{SetupBank}(\ell)$: Key generation for Bank B.
- $(\sigma_\Psi, d_\Psi, s_\Psi)() \leftarrow \mathsf{Withdraw}(\sigma_U; sk_{\mathsf{U}}, pk_{\mathsf{B}})(sk_{\mathsf{B}}, pk_{\mathsf{I}})$:
  User U withdraws an unspent e-coin $(\sigma_\Psi, d_\Psi, s_\Psi)$ from Bank B. The bank signs U's identity sk_{U} certified by Issuer I in signature σ_U.
- $(T, R)(\Psi) \leftarrow \mathsf{Spend}(\sigma_U, (\sigma_\Psi, d_\Psi, s_\Psi); sk_{\mathsf{U}})(pk_{\mathsf{B}})$:
  User U spends an e-coin $(\sigma_\Psi, d_\Psi, s_\Psi)$ with a recipient while proving ownership of the e-coin with relation to its identity sk_{U}. The recipient outputs a spent e-coin Ψ, whereas user U outputs (T, R) as auxiliary data to combine the $\mathsf{Spend}()$ method with other proofs. T is a function of $(sk_{\mathsf{U}}, R, d_\Psi)$, such as $T \leftarrow g_B^{sk_{\mathsf{U}}\mathrm{R}} g_B^{d_\Psi}$.
- $()(\Psi) \leftarrow \mathsf{Deposit}(\Psi)()$: Sends a spent e-coin Ψ to the bank.
- $(pk_{\mathsf{U}}, \Pi) \leftarrow \mathsf{Identify}(\Psi_1, \Psi_2)$:
  Bank B runs $\mathsf{Identify}()$ on two spent coins Ψ_1 and Ψ_2 to identify a double-spend perpetrator. It outputs a public key pk_{U} and a double-spending proof Π.
- 0 or $1 \leftarrow \mathsf{VerifyGuilt}(pk_{\mathsf{U}}, \Pi)$:
  The double-spend case (pk_{U}, Π) is publicly verifiable by $\mathsf{VerifyGuilt}()$.

Let us recall the core properties of an e-coin scheme:

Correctness. The Withdraw() and Spend() operations terminate successfully with honest participants. An honest recipient accepts a e-coin from a successful Spend().

Balance. No more e-coins can be spent than have been withdrawn.

Identification of Double-Spenders. Suppose bank B is honest. Let us consider U and W as honest users, who each of them received an e-coin during an execution of the Spend() protocol with an adversary, say $\Psi_1 = (s_\Psi, R_1, T_1, \Phi_1)$ and $\Psi_2 = (s_\Psi, R_2, T_2, \Phi_2)$. Then the adversary can be identified by the double-spending detection Identify() with overwhelming probability.

Public Key Recovery. The double-spending detection identifies a perpetrator U by outputting pk_{U}. We do not require full tracing as proposed in e-cash schemes.

Exculpability. Guilt in double-spending is publicly verifiable.

3 Privacy-friendly Incentives

Wikipedia provides documents to its users contributed by members of the community. This user-generated content varies in quality and can be significantly improved by (expert) reviews and comments. As most scientists know, good reviews are time-consuming, that is, come at a cost. Even though community service out of idealism is a common trait in the Wikipedia community, incentive systems can improve the situation for contributors as well as for the contributed content. They aim at reimbursing the review or revision cost by awards and at invigorating the review process.

Privacy-friendly incentives complement this fundamental goal with anonymity and privacy protection for all users. Therefore, they enable a double-blind peer review process and nurture fairness, impartiality, and rigor. Authors as well as the reviewers of documents can remain anonymous during the entire review process. Such a review process is believed to be essential for academic quality, even though it sometimes lacks in reviewer accountability. Our goal is to establish a cryptographic system that reaches high quality standards, while fulfilling the diverse requirements of the involved parties.

We formalize the incentive system as a collaborative document editing system, in which all revisions, reviews and comments are linked to one initial Document P_0. We consider a document version history $\mathbb{P} = \{P_0, \ldots P_n\}$ as ordered sequence of revisions, reviews and comments associated with the P_0, where P_n denotes the most recent revision or review.

Principals. There are multiple parties interacting with a document P. We have a clearing house that hosts all documents and organizes the incentive system, in our case a Wiki W. The Wiki has a community of users U. Each user may act in different and multiple roles:

Reader U: A reader consumes Document P. Any reader may offer incentives to other users to improve the quality of a document by a review or a revision.

Author V: An author contributes an initial version or a revision of a Document P.

Reviewer R: A reviewer contributes reviews and comments for a Document P in exchange for receiving an incentive.

Editor E: An editor is a privileged user, who may approve or decline document revisions or reviews by authors and reviewers.

We introduce a bank B to exchange real-world goods and awards for electronic incentives. Users of Wiki W can withdraw fresh incentive e-coins and deposit spent ones as part of our virtual incentive economy. Even though we allow a system with full anonymity, we require the users to register with a trusted identity Issuer I to infuse

accountability in the entire review and incentive process. Each User U obtains an identity certificate σ_{U} on its identity sk_{U} from Issuer I.[1] The identity of honest users is never revealed by the incentive system, whereas the certified identity enforces separation of duty between authors and reviewers and prevents double-spending attacks as well as vandalism.

Concepts. In a privacy-friendly incentive system, many anonymous users interact with a single Document P. Incentives may be given before or after a contribution (revision or review). *Pre-contribution* incentives are offered to users to provide a contribution at all and independent from the contribution quality. For instance, a Reader U can offer incentive e-coins for any reviewer who is willing to contribute a review. *Post-contribution* incentives are offered after the contribution is made and may be dependent on the quality of contribution. For instance, users can rate the quality of a reviewer's contribution and offer reputation coins for his work.

In our model, a Reader U explicitly withdraws incentives from a Bank B. The reader offers these *pre-contribution* incentives on the Wiki W for improvements to a document P. The Wiki W acts as clearing house and is responsible for ensuring unlinkability by exchanging the spent incentives of Reader U with Bank B for fresh incentives. Once a reviewer R decides to contribute a review P', he submits the review to the Wiki W for inspection by the editor. Once the editor approved the review, the Reviewer R can obtain the incentives from the Wiki W. We leave a community approval to the extensions in Section 5. As *post-contribution* incentives extension, the number of the obtained incentives can be dependent on the review rating or the reviewer can obtain separate reputation coins to build a reputation credential.

Checks and Balances. The privacy-friendly incentive system provides anonymity to all users and balances this property with strong accountability safe-guards. In a fully anonymous system without such safe-guards, malicious users could attempt to manipulate reviews, sabotage other author's work or publish fabrications without accountability. Well known examples of checks and balances to counter those attacks are the separation of reviewer and author/editor or the binding of reviews and documents to the contributor's true identity.

To achieve accountability as well as separation of duty between roles, we introduce a cryptographic domain pseudonym $N_{P,\mathsf{U}}$ for each User U that interacts with a Document P. It is function of the user's true identity sk_{U} and the page P while hiding sk_{U} computationally. Therefore, each principal interacting with P has one unique pseudonym, which is independent from their role. Pseudonyms $N_{P,\mathsf{U}}$ and $N_{Q,\mathsf{U}}$ from different documents P and Q are unlinkable.

4 Core Incentive System

4.1 Service Interface

- $()(N_{\mathsf{B},\mathsf{U}}) \leftarrow \mathsf{Register}(\sigma_{\mathsf{U}}; sk_{\mathsf{U}}, pk_{\mathsf{B}})(pk_{\mathsf{I}})$:
  A User U registers at Bank B anonymously while establishing a bank-specific domain pseudonym $N_{\mathsf{B},\mathsf{U}}$ for future transactions.

- $(\sigma_\Psi, d_\Psi, s_\Psi)(N_{\mathsf{B},\mathsf{U}}) \leftarrow \mathsf{WithdrawIncentive}(\sigma_{\mathsf{U}}; sk_{\mathsf{U}}, pk_{\mathsf{B}})(sk_{\mathsf{B}}, pk_{\mathsf{I}})$:
  A Reader U withdraws incentive coins from Bank B. Reader U outputs a triple of a coin signature σ_Ψ, a double-spend random element d_Ψ and a coin serial number s_Ψ. Bank B outputs the reader's domain pseudonym $N_{\mathsf{B},\mathsf{U}}$.

- $()(\Psi, N_{P,\mathsf{U}}) \leftarrow \mathsf{SubmitOffer}(\sigma_{\mathsf{U}}, (\sigma_\Psi, d_\Psi, s_\Psi), P; sk_{\mathsf{U}})(pk_{\mathsf{B}}, pk_{\mathsf{I}})$:
  A Reader U submits an incentive offer to the Wiki W, the clearing house. Wiki W ouputs a spent coin Ψ and the reader's domain pseudonym for Page P: $N_{P,\mathsf{U}}$. $\mathsf{SubmitOffer}$ guarantees the reader's proof of possession of the e-coin.

[1] Even our system works with multiple banks as well as multiple identity issuers, we focus on the single-bank/single-issuer case for simplicity.

- $()(N_{P,\mathsf{R}}) \leftarrow \mathsf{ProposeReview}(\sigma_{\mathsf{R}}, P, P'; sk_{\mathsf{R}}, pk_{\mathsf{I}})(\mathsf{Review}_P; pk_{\mathsf{I}})$:
  A Reviewer R proposes a Review P' for Document P anonymously at Wiki W. The Wiki outputs the reviewer's domain pseudonym $N_{P,\mathsf{R}}$ for Page P. It ensures that Reviewer R fulfills the entitlement and qualification predicate Review_P and the separation of duty with the author.

- $()(N_{P,\mathsf{E}}) \leftarrow \mathsf{EvaluateReview}(\sigma_{\mathsf{E}}, P, P', result; sk_{\mathsf{E}}, pk_{\mathsf{I}})(pk_{\mathsf{I}})$:
  Editor E rates the Review P' for Document P with rate $result$. The value $result$ determines approval or rejection. Wiki W enforces separation of duty.

- $(\Psi)(N_{P,\mathsf{R}}) \leftarrow \mathsf{SubmitReview}(\sigma_{\mathsf{R}}, P, P'; sk_{\mathsf{R}}, pk_{\mathsf{B}}, pk_{\mathsf{I}})(\sigma_{\mathsf{W}}, (\sigma_\Psi, d_\Psi, s_\Psi); sk_{\mathsf{W}}, pk_{\mathsf{I}})$:
  A Reviewer R submits an approved review P' to Wiki W and obtains the reward incentive coin Ψ in return. The domain pseudonym $N_{P,\mathsf{R}}$ links the transactions.

- $()(\Psi) \leftarrow \mathsf{DepositIncentive}(\Psi)()$: A spent coin Ψ is sent to Bank B.

- $(\sigma'_\Psi, d'_\Psi, s'_\Psi)(\Psi) \leftarrow \mathsf{ExchangeIncentive}(\Psi, \sigma_{\mathsf{W}}; sk_{\mathsf{W}}, pk_{\mathsf{B}})(sk_{\mathsf{B}}, pk_{\mathsf{I}})$:
  We introduce an $\mathsf{ExchangeIncentive}()$ method to allow Wiki W to deposit a spent coin Ψ at Bank B in exchange for a fresh coin $(\sigma'_\Psi, d'_\Psi, s'_\Psi)$. The Bank learns the deposited coin Ψ and may run $\mathsf{Identify}()$ to reveal double-spenders.

4.2 Requirements

First, to ensure rigorous and impartial reviews, authors, reviewers and editors must benefit from a strong privacy protection. The parties need to be anonymous and their transactions unlinkable between multiple documents.

Second, we consider multiple access control properties. The system must support roles and attributes to qualify reviews. We also allow the certification of reviewer profession and expertise, which increases trust in reviews and may entitle to claiming a larger incentive for an editing task. In addition, the roles of different parties in a review process must be clearly separated. The most common example is that an author may not review and judge her own article.

Third, the system must hold users accountable for their actions to discourage vandalism and fraud. This involves a certification of the user's identities, be it by the Wikipedia system itself or trusted third parties, such as government-supported eID issuers. The system supports identity escrow by standard means, for instance, by verifiably encrypting the user's true identity to a trusted anonymity revocation authority.

Incentive Security. *Correctness*: The operations terminate successfully with honest participants. *Balance*: No more incentive e-coins can be given than have been withdrawn. (a) An adversary can be identified by the double-spending detection with overwhelming probability. (b) Public Key Recovery identifies a perpetrator U by outputting pk_{U}. *Exculpability*: Double-spending guilt is publicly verifiable.

Anonymity. The users of a Wikipedia system can be completely anonymous. Users shall only be linked to specific articles by domain pseudonyms.

Unlinkability. Different transactions within a review process as well as transactions of the entire Wikipedia system are unlinkable. Unlinkability of underlying technology is orthogonal to this claim (IP addresses, cookies, etc.)

Role- and Attribute-based Entitlement. The system allows for role-based and attribute-based access control (RBAC/ABAC) based on certified identities.

Separation of Duty. The system enforces a separation of conflicting duties (SoD). In particular, an author cannot review or rate her own article.

Accountability. The system holds users accountable for their actions by three means: (i) *Identity Certification*: The users' true identities and roles are certified in anonymous credentials by trusted issuers. (ii) *Master Key Consistency*: All credentials and transactions of a user U are all bound to the same identity/master key sk_{U}. (iii) *Identity Escrow*: We allow a trusted third party to revoke the anonymity of users.

4.3 Realization

We realize the incentive service interface with the abstract primitives from Section 2.

$\mathsf{Register}(\sigma_{\mathsf{U}}; sk_{\mathsf{U}}, pk_{\mathsf{B}})()$**.** We require each User U to register at Bank B and to establish a bank-specific domain pseudonym $N_{\mathsf{B,U}}$ in the course of action. User U proves knowledge of representation of the domain pseudonym in SPK_1:

$$
\begin{aligned}
SPK_1\{&(\sigma_{\mathsf{U}}, sk_{\mathsf{U}}, m_1, \ldots, m_l) : \\
&\mathsf{VerifySign}(\sigma_{\mathsf{U}}, (m_1, \ldots, m_l); pk_{\mathsf{I}}) \wedge \\
&N_{\mathsf{B,U}} = (\mathcal{H}(pk_{\mathsf{B}}))^{sk_{\mathsf{U}}} \\
\};&
\end{aligned}
$$

$$()(N_{\mathsf{B,U}}) \leftarrow \mathsf{Register}(\sigma_{\mathsf{U}}; sk_{\mathsf{U}}, pk_{\mathsf{B}})()$$

User U $(\sigma_{\mathsf{U}}; sk_{\mathsf{U}}, pk_{\mathsf{B}})$		Bank B (pk_{I})
$N_{\mathsf{B,U}} \leftarrow (\mathcal{H}(pk_{\mathsf{B}}))^{sk_{\mathsf{U}}}$;		
$\Phi_1 \leftarrow SPK_1\{\ldots\}$	$\xrightarrow{(N_{\mathsf{B,U}}, \Phi_1)}$	Verify: $(N_{\mathsf{B,U}}, \Phi_1)$ with pk_{I}
$()$		$(N_{\mathsf{B,U}})$

$\mathsf{WithdrawIncentive}(\sigma_{\mathsf{U}}; sk_{\mathsf{U}}, pk_{\mathsf{B}})(sk_{\mathsf{B}}, pk_{\mathsf{I}})$**.** We require a Reader U withdrawing an incentive e-coin to prove her pseudonym $N_{\mathsf{B,U}}$ prior to the coin withdraw with SPK_2:

$$
\begin{aligned}
SPK_2\{&(\sigma_{\mathsf{U}}, sk_{\mathsf{U}}, m_1, \ldots, m_l) : \\
&\mathsf{VerifySign}(\sigma_{\mathsf{U}}, (m_1, \ldots, m_l); pk_{\mathsf{I}}) \wedge \\
&N_{P,\mathsf{U}} = (\mathcal{H}(pk_{\mathsf{B}}))^{sk_{\mathsf{U}}} \\
\};&
\end{aligned}
$$

After the Reader U successfully logged in as $N_{\mathsf{B,U}}$, it engages in a $\mathsf{Withdraw}()$ operation with the bank, to obtain the incentive e-coins.

$$(\sigma_\Psi, d_\Psi, s_\Psi)(N_{\mathsf{B},\mathsf{U}}) \leftarrow \mathsf{WithdrawIncentive}(\sigma_\mathsf{U}; sk_\mathsf{U}, pk_\mathsf{B})(sk_\mathsf{B}, pk_\mathsf{I})$$

Reader U $(\sigma_\mathsf{U}; sk_\mathsf{U}, pk_\mathsf{B})$		Bank B $(sk_\mathsf{B}, pk_\mathsf{I})$
$N_{\mathsf{B},\mathsf{U}} \leftarrow (\mathcal{H}(pk_\mathsf{B}))^{sk_\mathsf{U}}$;		
$\Phi_2 \leftarrow SPK_2\{\ldots\}$	$\xrightarrow{(N_{\mathsf{B},\mathsf{U}}, \Phi_2)}$	Verify: $(N_{\mathsf{B},\mathsf{U}}, \Phi_2)$ with pk_I
$(\sigma_\Psi, d_\Psi, s_\Psi)$	$\xleftarrow{\mathsf{Withdraw}(\sigma_\mathsf{U}; sk_\mathsf{U}, pk_\mathsf{B})(sk_\mathsf{B}, pk_\mathsf{I})}$	$()$
$(\sigma_\Psi, d_\Psi, s_\Psi)$		$(N_{\mathsf{B},\mathsf{U}})$

$\mathsf{SubmitOffer}(\sigma_\mathsf{U}, (\sigma_\Psi, d_\Psi, s_\Psi), P; sk_\mathsf{U})(pk_\mathsf{B})$**.** To submit an offer, a Reader U spends an incentive e-coin with the Wiki and proves knowledge of representation of his domain pseudonym at a document P by SPK_3:

$$\begin{aligned} SPK_3\{(sk_\mathsf{U}, d_\Psi, R) :& \\ & T = g_B^{sk_\mathsf{U}\mathsf{R}} g_B^{d_\Psi} \wedge \\ & N_{P,\mathsf{U}} = (\mathcal{H}(P))^{sk_\mathsf{U}} \\ \};& \end{aligned}$$

$$()(\Psi, N_{P,\mathsf{U}}) \leftarrow \mathsf{SubmitOffer}(\sigma_\mathsf{U}, (\sigma_\Psi, d_\Psi, s_\Psi), P; sk_\mathsf{U})(pk_\mathsf{B})$$

Reader U $(\sigma_\mathsf{U}, (\sigma_\Psi, d_\Psi, s_\Psi), P; sk_\mathsf{U})$		Wiki W $(pk_\mathsf{B}, pk_\mathsf{I})$
$N_{P,\mathsf{U}} \leftarrow (\mathcal{H}(P))^{sk_\mathsf{U}}$;		
(T, R)	$\xleftarrow{\mathsf{Spend}(\sigma_\mathsf{U}, (\sigma_\Psi, d_\Psi, s_\Psi); sk_\mathsf{U})(pk_\mathsf{B}, pk_\mathsf{I});}$	(Ψ)
$\Phi_3 \leftarrow SPK_3\{\ldots\}$	$\xrightarrow{(\Phi_3)}$	Verify: Φ_3 with pk_I
$()$		$(\Psi, N_{P,\mathsf{U}})$

Note that even though this protocol does not provide inherent fairness (each party may abort between $\mathsf{Spend}()$ and SPK_3), none of the parties may gain an advantage by misbehaving.

$\mathsf{ProposeReview}(\sigma_\mathsf{R}, P, P'; sk_\mathsf{R}, pk_\mathsf{I})(\mathsf{Review}_P; pk_\mathsf{I})$**.** A Reviewer R may propose a Review P' for Document P by proving knowledge of representation of her domain pseudonym $N_{P,\mathsf{R}}$ to Wiki W. SPK_4 proves in addition that the Reviewer certificate σ_R fulfills the predicate Review_P, e.g., that Reviewer R is a doctor:

$$\begin{aligned} SPK_4\{(\sigma_\mathsf{R}, sk_\mathsf{R}, m_1, \ldots, m_l) :& \\ & \mathsf{VerifySignPred}(\sigma_\mathsf{R}, (m_1, \ldots, m_l), \mathsf{Review}_P; pk_\mathsf{I}) \wedge \\ & N_{P,\mathsf{R}} = (\mathcal{H}(P))^{sk_\mathsf{R}} \\ \};& \end{aligned}$$

The Wiki verifies that the Reviewer R is unequal of the document's author by comparing their domain pseudonyms.

$$()(N_{P,\mathsf{R}}) \leftarrow \mathsf{ProposeReview}(\sigma_\mathsf{R}, P, P'; sk_\mathsf{R}, pk_\mathsf{I})(\mathsf{Review}_P; pk_\mathsf{I})$$

Reviewer R Input: $(\sigma_\mathsf{R}, P, P'; sk_\mathsf{R}, pk_\mathsf{I})$		Wiki W Input: $(\mathsf{Review}_P; pk_\mathsf{I})$
$N_{P,\mathsf{R}} \leftarrow (\mathcal{H}(P))^{sk_\mathsf{R}}$; $\Phi_4 \leftarrow SPK_4\{\ldots\}$	$\xrightarrow{((N_{P,\mathsf{R}}, \Phi_4)}$	Verify: $N_{P,\mathsf{R}} \neq N_{P,\mathsf{V}}, \Phi_4$ with pk_I
$()$		$(N_{P,\mathsf{R}})$

$\mathsf{EvaluateReview}(\sigma_\mathsf{E}, P, P', result; sk_\mathsf{E}, pk_\mathsf{I})(pk_\mathsf{I})$**.** An Editor E can evaluate a review after having proven knowledge of representation of his domain pseudonym in SPK_5:

$$\begin{aligned} SPK_5\{&(\sigma_\mathsf{E}, sk_\mathsf{E}, m_1, \ldots, m_l) : \\ &\mathsf{VerifySign}(\sigma_\mathsf{E}, (m_1, \ldots, m_l); pk_\mathsf{I}) \wedge \\ &N_{P,\mathsf{E}} = (\mathcal{H}(P))^{sk_\mathsf{E}} \\ \}&(result); \end{aligned}$$

The Wiki verifies that Editor E and Reviewer R are unequal by comparing their domain pseudonyms.

$$()(N_{P,\mathsf{E}}, result) \leftarrow \mathsf{EvaluateReview}(\sigma_\mathsf{E}, P, P', result; sk_\mathsf{E}, pk_\mathsf{I})(pk_\mathsf{I})$$

Editor E $(\sigma_\mathsf{E}, P, P', result; sk_\mathsf{E}, pk_\mathsf{I})$		Wiki W (pk_I)
$N_{P,\mathsf{E}} \leftarrow (\mathcal{H}(P))^{sk_\mathsf{E}}$; $\Phi_5 \leftarrow SPK_5\{\ldots\}$	$\xrightarrow{(N_{P,\mathsf{E}}, result, \Phi)}$	Verify: $N_{P,\mathsf{R}} \neq N_{P,\mathsf{E}}, \Phi(result)$ with pk_I
$()$		$(N_{P,\mathsf{E}}, result)$

$\mathsf{SubmitReview}(\sigma_\mathsf{R}, P, P'; sk_\mathsf{R}, pk_\mathsf{B}, pk_\mathsf{I})(\sigma_\mathsf{W}, (\sigma_\Psi, d_\Psi, s_\Psi); sk_\mathsf{W}, pk_\mathsf{I})$**.** When a Reviewer R submits an approved review P', she needs to prove knowledge of representation of her domain pseudonym first to link the transaction to the previous ones:

$$\begin{aligned} SPK_6\{&(\sigma_\mathsf{R}, sk_\mathsf{R}, m_1, \ldots, m_l) : \\ &\mathsf{VerifySign}(\sigma_\mathsf{R}, (m_1, \ldots, m_l); pk_\mathsf{I}) \wedge \\ &N_{P,\mathsf{R}} = (\mathcal{H}(P))^{sk_\mathsf{R}} \\ \}&; \end{aligned}$$

Wiki W only engages in a $\mathsf{Spend}()$ protocol run with Reviewer R after a successful proof. By that, the Reviewer R obtains an incentive coin Ψ and can subsequently deposit it with the bank.

$$(\Psi)(N_{P,\mathsf{R}}) \leftarrow \mathsf{SubmitReview}(\sigma_\mathsf{R}, P, P'; sk_\mathsf{R}, pk_\mathsf{B}, pk_\mathsf{I})(\sigma_\mathsf{W}, (\sigma_\Psi, d_\Psi, s_\Psi); sk_\mathsf{W}, pk_\mathsf{I})$$

Reviewer R $(\sigma_\mathsf{R}, P, P'; sk_\mathsf{R}, pk_\mathsf{B}, pk_\mathsf{I})$		Wiki W $(\sigma_\mathsf{W}, (\sigma_\Psi, d_\Psi, s_\Psi); sk_\mathsf{W}, pk_\mathsf{I})$
$N_{P,\mathsf{R}} \leftarrow (\mathcal{H}(P))^{sk_\mathsf{R}};$		
$\Phi_6 \leftarrow SPK_6\{\ldots\}$	$\xrightarrow{(N_{P,\mathsf{R}}, \Phi_6)}$	Verify: $(N_{P,\mathsf{R}}, \Phi_6)$ with pk_I
(Ψ)	$\xleftarrow{\mathsf{Spend}(\sigma_\mathsf{W}, (\sigma_\Psi, d_\Psi, s_\Psi); sk_\mathsf{W})(pk_\mathsf{B})}$	(T, R)
(Ψ)		$(N_{P,\mathsf{R}})$

4.4 Security Analysis

Incentive Security. The balance property of the e-cash system directly transfer to the incentive balance of our construction. The e-cash system's Identify() and VerifyGuilt() operations on user public keys enforce *Balance* and *Exculpability*.

Anonymity and Unlinkability. We found our construction on anonymous credentials as root identity. Throughout the system's transactions, users only prove knowledge of representation of their domain pseudonyms and keep their actual identities confidential.[2]

The cross-document unlinkability is maintained because the domain pseudonyms are uniformly distributed random group elements under the assumption of the random oracle model (ROM). The decision whether two keys x and y are equal given $(\mathcal{H}(P))^x, \mathcal{H}(Q))^y, P, Q)$ is hard under the DDH assumption in the ROM. We break the linking of non-transferable e-coins by an exchange[3] between clearing house and bank.[4]

Role- and Attribute-based Entitlement. We use the certified attributes in a user's identity credential σ_U as flexible entitlement mechanism. Our system supports RBAC by certified role attributes as well as ABAC by selective disclosure of further attributes. We employ this technique in the ProposeReview() operation: a Reviewer R proves that her identity credential σ_R fulfills a review predicate Review_P. It applies to all proofs of representation of domain pseudonyms.

Separation of Duty. The separation of duty is enforced by proofs over domain pseudonyms. Whereas we achieve a subject-based separation of duty[5] by an inequality check of the domain pseudonyms, we realize role-based separation of duty[6] with signature proofs of knowledge of roles and attributes associated with a domain pseudonym. The probability that two keys x and y for $(\mathcal{H}(P))^x, \mathcal{H}(P))^y)$ collide is negligible as both are uniformly distributed random group elements given the ROM.

Accountability. We get *Certification* and *Consistency* properties by design of the anonymous credential system. We achieve *Identity Escrow* by including a Verifiable Encryption (e.g., Camenisch and Shoup [13]) of a user's true identity sk_U towards a trusted third party. We consider this procedure a standard technique and do not elaborate on it.

[2] Only exception is the double-spending detection.

[3] ExchangeIncentive() chains DepositIncentive() and WithdrawIncentive()

[4] Clearly, the linking through underlying network channels (e.g., IP and MAC addresses) is orthogonal to the presented scheme.

[5] "The editor of the document is unequal to the author."

[6] "The document can only be confirmed under four-eyes principle, where one user has the role *Clerk* and another has the role *Manager*."

5 Functional Extensions and Future Work

Whereas we implemented the core incentive system as elaborated in Section 6, we have not realized certain extension ideas, yet.

5.1 Rating Reviews

In the presented system, the Wiki is responsible for checking the quality of the reviews and, if it finds the quality sufficient, for releasing spent incentive coins to the reviewer. Alternatively, one could let the Wiki community rate these reviews and have the Wiki only release the coins to the reviewer, if the submitted review obtained a sufficiently high ranking. We proceed as follows: Users (raters) sign their rating of the review with their domain pseudonym. The Wiki collects these ratings, checks that the domain pseudonym of the raters and the reviewer are unequal (separation of duty) as well as that each domain pseudonym of a rater only occurs once (one-time rating).

If there are several reviews, the offered coins can be distributed to different reviewers in proportion of the ranking. This approach encourages reviewers to provide quality reviews in order to gain a high ranking and collect most of the coins. At the same time, it prevents reviewers from collecting all of the coins for poor quality reviews.

5.2 Reviewer Reputation

A rating of a review provides a feedback on the quality of the review. This naturally lends itself to be used for an (anonymous) reputation system. Thus, the Wiki could issue reputation credentials (points) to reviewers and authors based on the quality of the reviews and articles.[7]

More precisely, a reputation system can be implemented as follows. In addition to earned incentives, the Wiki could also issue an anonymous one-time credential to the user (reviewer or author) according to the received average rating. This credential can be realized with the e-coin scheme, where the rating is encoded in the denomination. One either uses a different Bank public key for each denomination or one extends the e-cash scheme to include denomination as an attribute of an e-coin. These *reputation e-coins* can then b e gathered by the author or reviewers. Let us assume some reputation authority that issues credentials which state user's reputation. Then, users can then exchange the reputation e-coins with the reputation authority against an updated reputation credential without this transaction being linkable to the corresponding article/review. The one-time spending property of the e-coin will ensure that each rating can only be used only once. Depending on how the reputation is computed, the rating e-coin and the old reputation credential cannot be exchanged directly for a new reputation credential, but the user might need to have a pseudonymous account where he can deposit all the different ratings and then get an updated reputation credential issued once this computation is done. We leave a detailed discussion to the extended version of this paper.

Each user U holds a reputation credential $\sigma_{U,\mathsf{Rep}}$ by the Bank BR, which certifies the current reputation attribute associated with sk_{U}. $\mathsf{SubmitReputationOffer}()$ is identical to $\mathsf{SubmitOffer}$, with the exception that the Wiki W verifies $N_{P,\mathsf{U}} \neq N_{P,\mathsf{R}}$. $\mathsf{AcceptReputationOffer}()$ is straight-forward. $\mathsf{DepositReputationIncentive}()$ proves the current reputation of $\sigma_{U,\mathsf{Rep}}$, deposits received reputation coins and obtains a new reputation credential $\sigma'_{U,\mathsf{Rep}}$.

$(\sigma'_{U,\mathsf{Rep}})(N_{\mathsf{BR,U}}, \Psi) \leftarrow \mathsf{DepositReputationIncentive}(\sigma_U, \sigma'_{U,\mathsf{Rep}}, \Psi; sk_{\mathsf{U}}, pk_{\mathsf{BR}})(sk_{\mathsf{BR}}, pk_{\mathsf{I}})$:

- $r \xleftarrow{\$} \mathbb{Z}_q$;
- $C \leftarrow \mathsf{Commit}(sk_{\mathsf{U}}, r)$;
- $N_{\mathsf{BR,U}} \leftarrow (\mathcal{H}(pk_{\mathsf{BR}}))^{sk_{\mathsf{U}}}$;

[7] Articles could be ranked by users similarly to the reviews as described above.

- $SPK\{(\sigma_{\mathsf{U}}, \sigma_{\mathsf{U,Rep}}, sk_{\mathsf{U}}, m_1, \ldots, m_l, rep, r) :$
  $\mathsf{VerifySign}(\sigma_{\mathsf{U}}, (m_1, \ldots, m_l); pk_{\mathsf{I}}) \wedge$
  $\mathsf{VerifySign}(\sigma_{\mathsf{U,Rep}}, rep; pk_{\mathsf{I}}) \wedge$
  $\mathsf{VerifyCommit}(C, sk_{\mathsf{U}}, r) \wedge$
  $N_{\mathsf{BR,U}} = (\mathcal{H}(pk_{\mathsf{BR}}))^{sk_{\mathsf{U}}}$
  $\};$
- $(\sigma'_{U,\mathsf{Rep}})() \leftarrow \mathsf{HiddenSign}((C), (rep'), r; pk_{\mathsf{BR}})(sk_{\mathsf{BR}});$
- Output: $(\sigma'_{U,\mathsf{Rep}})(N_{\mathsf{BR,U}});$

6 Example application

Wikipedia is a large-scale online encyclopedia project that at this time has grown to $\sim 3.2 \cdot 10^6$ articles in the English version ($\sim 12.5 \cdot 10^6$ articles in total), and that is beginning to rival more established compendiums of human knowledge [19]. Its software platform, MediaWiki, allows anybody with Internet access to read and edit shared articles. The most important criteria in Wikipedia's search for new quality assessment methods are the immediacy typical for social media as well as accuracy, which can be challenging [25]. The German Wikipedia chapter has deployed one such process in the form of the MediaWiki extension `FlaggedRevs`[8]. It allows eligible users[9] to review articles. In one configuration, review criteria include levels of accuracy, depth, and readability, and the review status is prominently displayed along with each article as important quality indication. According to the 2008 report, approximately 90.8% of the German Wikipedia articles have been reviewed at least once, even though, mostly by small pockets of active (expert) contributors.

Our example application functions as an extension to the `FlaggedRevs` extension and is registered as add-on PHP functionality in MediaWiki. We register several incentive handling functions at its main code entry points. If both extensions are installed, users can offer incentives when they want certain articles to be reviewed, and reviewers can earn these incentives by providing their expert insights through the revision process. The last reviewed version of an article is referred to as its *stable* version (see Figure 1), others as non-stable. Figure 1 depicts state transitions per article with respect to reviews.

New or edited articles arrive in state 1. Users with reviewer status see articles with additional user interface elements that allow them to affect state transitions marked "Submit review". Other users see articles with additional user interface elements that allow them to affect state transitions marked "Submit offer" (see also Figure 2). In comparison, the base MediaWiki platform supports only state 3, MediaWiki with `FlaggedRevs` supports only states 1 and 3, but not yet 2.

We report that we have implemented the presented incentive system architecture in MediaWiki (see e.g. Figure 3) as an additional extension to MediaWiki, as well as the cryptographic incentives system from Section 3 based on the Identity Mixer cryptographic library [20], using the SRSA setting. The MediaWiki extension contains the appropriate hooks to accommodate the cryptographic functions and serves as a glue between MediaWiki and the incentives system.

6.1 System architecture

6.1.1 Static design.

Figure 2 shows the same principals as defined in Section 3 and concentrates on the software architecture of components that allow humans to participate in the scheme. High-level components have been realized as Java servlets.

[8] http://preview.tinyurl.com/davsm4

[9] e.g. those who have earned editor role by being active community member for a certain duration.

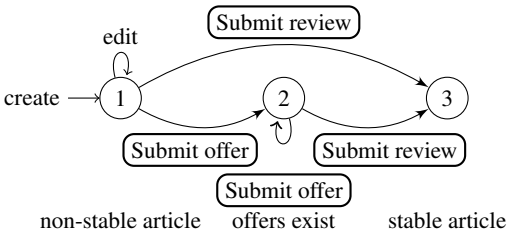


Figure 1: State transitions during reviews.

Appendix Section E lists their configuration parameters. Each user-facing component has an id, a password (corresponding to its MediaWiki account information) and a pair of cryptographic keys (for participating in the protocols). The bank B_0 and the clearing W are special in that certain other components must have knowledge of these principals' public keys and network addresses in their configurations.

The components marked as *user* and *reviewer* correspond to U and R as defined in Section 3, and they receive communication at anonymous network addresses addr_{U} and addr_{R} respectively. Internally, the two are exactly the same, and any principal who controls a component of this kind can participate in the sample application either as a reviewer or other user (the distinction depending only on her current reviewer status in MediaWiki). The component marked as *clearing* corresponds to W, it receives communication at address addr_{W}. The clearing W functions as a front-end to MediaWiki and its extensions, thus linking its core logic (implemented in PHP and JavaScript by MediaWiki conventions) to privacy-friendly incentives system (implemented in Java). The component marked *bank* corresponds to B_0 and it receives communication at address $\text{addr}_{\mathsf{B}_0}$.

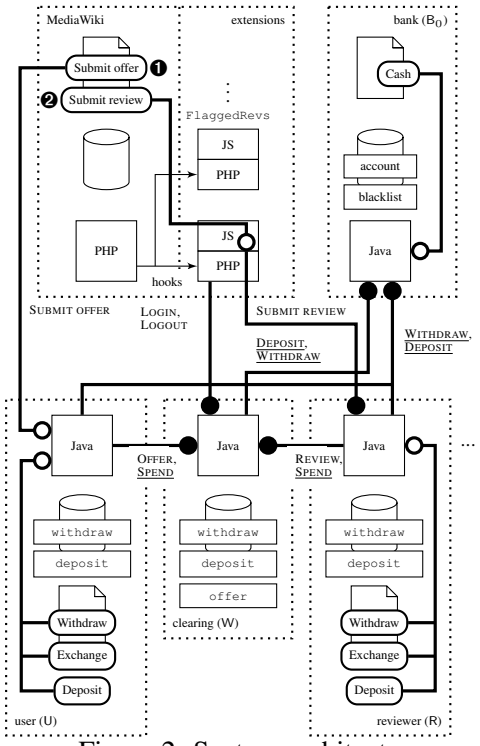


Figure 2: System architecture.

All principal components maintain relational databases (cylinder shapes in Figure 2) locally. The architecture does not assume that their private data is stored at any central location. The individual tables contain the following information.[10] Each bank maintains two tables: `account` is a mapping $\text{id}_U \mapsto (\text{pk}_U, n)$, where id_U is domain pseudonym computed by user using address of the bank and n is the current balance of user U's account at the bank. `blacklist` is a mapping $\text{id}_U \mapsto \{ x_j \}_j$, where x_j are textual log entries pertaining to past double-spending behavior by user U, including the proof of double spending which can be verified by other parties. Each user maintains two tables: `withdraw` is a set of coins $\{ \Psi_i \}_i$ that were withdrawn from a bank and have not yet been spent. `deposit` is a set of spent coins $\{ \Omega_i \}_i$ that have been received from another user, but have not yet been deposited at a bank. The clearing is an extension of the user component and maintains additional table: `offer` is a mapping $\text{article} \times \text{id}_U \mapsto (n)$ where n is the number of coins offered by user U for a review of the article.

The flow of control in the example application proceeds from user interactions in MediaWiki's web interface, and in additional web interfaces (document shapes with superimposed user interface elements in Figure 2) contributed by several components. The clearing component does not contribute a separate web interface, because it is an intrinsic part of the MediaWiki extension and therefore co-occupies MediaWiki's own user interface. Individual control elements are linked to the following specific actions with respect to privacy-friendly incentives. (The control elements "Submit offer" and "Submit review" are covered in more detail later in this

[10] For all indicated mappings, $a \mapsto b$ serves as shorthand notation for the mapping from the set of all possible values for a to the set of all possible values for b.

section.)

"Cash" is used by the bank principal to float the virtual currency system by setting values n in users' accounts. How the total circulating volume is controlled (e.g. by requiring backing by other instruments, such as credit cards accounts) is outside the scope of the present system. WITHDRAW allows users (both U and R) to obtain coins from a bank, provided that their respective accounts have sufficient coverage. DEPOSIT allows them to deposit received coins and thereby increase their account balance. The connections ending with empty disks in Figure 2 indicate submission of HTML forms whereby principals can initiate some primary action. The connections ending with solid disks indicate programmatic HTTP communications between components that follow from these primary actions. Protocols executed between the MediaWiki extension and the incentives system send XML payload over HTTP, and their design is inspired by the architectural style REST [18]. Disks (proximate to HTTP servers) indicate the direction of connection establishment. Protocols executed between different components of the incentives system use serialized Java objects for communication as the Idemix library does not yet support full XML serialization of its main objects used in these protocols. The future goal is to use XML payloads for all of the protocols as it reduces the dependency on a particular version of the serialized object.

6.1.2 Dynamic design.

We will now explain the dynamic aspects of the system by following two representative use cases. In the first use case a user offers privacy-friendly incentive points for an article review. The flow for this use case starts at the points marked ❶ in Figure 2 (a user presses the "Submit offer" button). Our walk-through assumes that an eligible user has already logged into MediaWiki (see Section 6.2) and that the system is in the right state (see Figure 1), so that the user will actually see and can press the control button.

Step 1.1 A user decides that she wishes to offer coins (units of privacy-enabled incentive points) for the review of a non-stable article. To do so, she fills in an HTML form (see Figure 3(a)) and presses the "Submit offer" button.

Step 1.2 The web browser submits her request to the user U component, which runs locally on user's machine in a form of a Java servlet. The U component checks it's `withdraw` table whether there is sufficient number of coins, if not it will contact the bank B_0 with user's consent and get additional coins executing the WITHDRAW protocol and continue with submitting the offer.

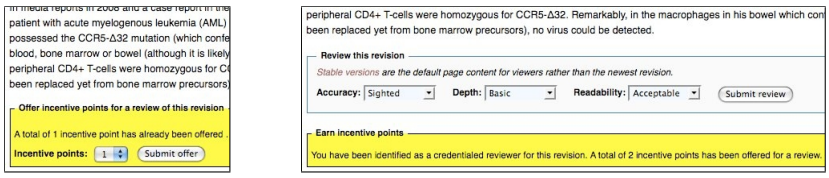
Step 1.3 The user U component contacts the clearing W component and passes a SUBMIT OFFER request which initiates the SPEND protocol and the offered coins are transferred, after being removed from the user U component table `withdraw`.

Step 1.4 On success, the clearing W interacts with the bank B_0 to exchange the coins (by executing DEPOSIT, then WITHDRAW) for fresh ones and stores them in its table `withdraw`, they will be later spent during a review. [11] Identification of the article to which the offer pertains along with number of coins offered are stored in the table `offer`.

In the second use case a reviewer receives privacy-friendly incentive points after conducting a review. The flow for this use case starts at the points marked ❷ in Figure 2 (a reviewer presses the button "Submit review"). Our walk-through assumes that an eligible reviewer has already logged into MediaWiki (see Section 6.2) and that the system is in the right state (see Figure 1), so that the reviewer will actually see and can press the control button.

Step 2.1 A reviewer R chooses to submit a review. To do so, she fills in HTML form (see Figure 3(b)) and presses the "Submit review" button.

[11] We note that a coin exchange independent from an actual review request protects against timing-based linking.



(a) Submitting an offer. (b) Earning incentives.

Figure 3: Our privacy-friendly incentives realization in use.

Step 2.2 This results in contacting the Java servlet of the reviewer R and an invocation of the clearing W. In order to satisfy the existing MediaWiki conventions (also those inherited from the `FlaggedRevs` extension) and JavaScript security rules that constrain remote communication, this invocation has to happen in a somewhat convoluted way involving both JavaScript and PHP components.

Step 2.3 When the clearing W finally receives the call, it looks up whether its table `offer` contains a fitting entry. On success, it spends the amount of offered coins out to reviewer R with the SPEND protocol while executing the SUBMIT REVIEW protocol. If this works out, it deletes the corresponding entry from `offer`.

Step 2.4 The reviewer R receives the coins and interacts with the bank B_0 to exchange the coins (by executing DEPOSIT, then WITHDRAW) for fresh ones and stores them in its table `withdraw`.

6.2 Anonymous and pseudonymous use

MediaWiki normally allows *pseudonymous use*, i.e. users do not have to reveal their real permanent identities but can operate under disguised permanent identities (false names) instead. While this affords a relatively large level of privacy, it is less than one wants to achieve with anonymous e-cash, as the name implies. *Anonymous use* occurs when users can operate under disguised transient identities (no names), meaning that two operations by the same user cannot be linked elsewhere to the same identity.

The MediaWiki system architecture supports pseudonymous use in that users can register under arbitrary user names. Yet all operations they conduct under those users names are carried out on a central server, which can therefore link them by keeping records.

Our privacy-friendly incentives system from section 3 can be extended to allow fully anonymous access to Wikipedia. In order to achieve this, the access control of Wikipedia needs to be adjusted to use domain pseudonym of a user and a proof that the user has properly registered, while retaining the original MediaWiki platform as is. When a user logs in under this design, she is not asked for her (pseudonymous) user name and password, but for an anonymous credential, anonymous network address, and password instead. This arrangement is a simulation of truly anonymous use, because at this time we do not yet present actual anonymous credentials (such as those from [8]) to MediaWiki, but rather use MediaWiki's own pseudonymous account names as simple substitutes for such credentials. (We refer to this as *semi-anonymous use* throughout this paper.) Semi-anonymous use has the advantage that we do not have to change MediaWiki's inner operation to make it aware of anonymous credentials: more straightforward use of MediaWiki's regular extension mechanism suffices instead to introduce the necessary changes. (It also causes that we still prompt the user for a password, although this would no longer be required with anonymous credentials.) If users were to show anonymous credentials to MediaWiki, they would do so for authentication at the start of communication sessions and MediaWiki would therefore learn also a transient address for anonymous communication with the user's component. This is not the case with semi-anonymous use. The simulation accounts for this by requiring that the user's (anonymous) communication endpoint address for a session is also explicitly specified during session login. In a production setting past the lifetime of this prototype the server would learn such addresses implicitly when it receives an anonymous credential from another party.

As is common for zero-knowledge-proof-based schemes, we assume that communication occur over anonymous channels (e.g. anonymous communication services for TCP-based traffic, such as [16]), so that endpoint addresses for succeeding communications are also unlinkable, even if communication arrives from the same (anonymous) user, provided that the senders' messages are spread over different routes. Our system architecture uses HTTP for all specific remote communication protocols (the connections ending with solid disks in figure 2), and suitably opaque URLs can serve as anonymous network addresses. For instance those could be formed from permanent versions by replacing the host part by an anonymous/numeric IP address. Notice that anonymous use with anonymous credential would not diminish the quality or discernment of expert opinions, because users, while anonymous, can still be well-credentialed.

7 Related work

To the best of our knowledge, we are the first to suggest a privacy-friendly incentive system for rating contributions in collaborative workspaces or p2p networks.

Incentives are useful to create reputation systems. Steinbrecher studied privacy-protecting reputation systems [24] using pseudonyms. In such pseudonymous solutions, the transactions that are taken into account to build reputation can all be linked together. Therefore, many authors have claimed that achieving privacy in reputation systems is impossible [22]. In contrast, in our scheme one can build reputation from different transactions without these being linkable. Adler and de Alfaro [2] propose an orthogonal content-driven trust extension for MediaWiki, called WikiTrust. They focus on the analysis of a document's author, her reputation, origin, and trust, whereas our system considers the users' interactions in a double-blind review system.

Lysyanskaya and co-authors [5] have proposed and implemented an incentive system based on plain e-cash and fair exchange or file sharing applications. Their work focuses on the (fair) exchange of token and digital items, whereas we are interested in the (anonymous) relationships of the parties receiving and offering the e-cash to enhance the quality of content. Their approach is orthogonal to ours.

8 Conclusion

This paper has introduced the novel concept of a privacy-friendly incentive system to rate user-generated content. We have proposed the first realization of such a system that draws on ideas from e-cash and anonymous credentials. The presented solution is privacy-friendly both from a theoretical and an applied perspective. In addition, we have contributed a practical architecture that integrates well with the open-source collaboration platform MediaWiki. To this end, we have extended MediaWiki for semi-anonymous use in a prototype environment, and designed the architecture such that it can support anonymous use by later adding anonymous credentials for authentication.

We report that we have implemented the cryptographic incentive system on top of the Identity Mixer library [9, 20]. We have realized a MediaWiki incentive extension with appropriate hooks for the cryptography and we integrated both results into MediaWiki, that is, hooking the cryptographic protocol implementation into the MediaWiki plug-ins. We believe that providing such an incentive system nurtures high-quality content on electronic collaboration platforms by vigorous user interaction and rigorous double-blind reviews. We hope for a raise in quality and trustworthiness of user-generated content, in particular, if earned incentive points can be exchanged (at a suitable exchange rate) into real goods, such as CDs or vouchers.

Finally, note that our solution can be extended in multiple ways, most prominently through: (i) multifaceted incentives, (ii) transferable e-cash, (iii) identity escrow, and (iv) complex roles and policies.

Acknowledgment

This work has been funded by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 216483.

References

[1] PrimeLife project, website. www.primelife.eu.

[2] B. T. Adler and L. de Alfaro. A content-driven reputation system for the Wikipedia. In *In Proceedings of the 16th International World Wide Web Conference (WWW) 2007*, pages 261–270, New York, NY, USA, 2007. ACM Press.

[3] M. H. Au, W. Susilo, and Y. Mu. Constant-size dynamic k-TAA. In R. D. Prisco and M. Yung, editors, *SCN 2006*, pages 111–125, 2006.

[4] E. Bangerter, J. Camenisch, and A. Lysyanskaya. *Network Security*, chapter A Cryptographic Framework for the Controlled Release Of Certified Data. Scott C.-H. Huang, David McCallum, and Ding-Zhu Du (Editors). To appear, 2006.

[5] M. Belenkiy, M. Chase, C. C. Erway, J. Jannotti, A. Küpçü, and A. Lysyanskaya. Incentivizing outsourced computation. In *NetEcon*, pages 85–90, 2008.

[6] D. Boneh and X. Boyen. Short signatures without random oracles. In C. Cachin and J. Camenisch, editors, *Advances in Cryptology — EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science* , pages 54–73. Springer, 2004.

[7] D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In M. K. Franklin, editor, *Advances in Cryptology — CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science* , pages 41–55. Springer Verlag, 2004.

[8] J. Camenisch and E. V. Herreweghen. Design and implementation of the idemix anonymous credential system. In *Proceedings of the 2002 ACM Conference on Computer and Communications Security, CCS 2002*, 2002.

[9] J. Camenisch and E. V. Herreweghen. Design and implementation of the *idemix* anonymous credential system. Technical Report Research Report RZ 3419, IBM Research Division, May 2002.

[10] J. Camenisch, S. Hohenberger, and A. Lysyanskaya. Compact E-cash. In R. Cramer, editor, *Advances in Cryptology — Eurocrypt 2005*, volume 3494 of *Lecture Notes in Computer Science* , pages 302–321. Springer, 2005.

[11] J. Camenisch, A. Kiayias, and M. Yung. On the portability of generalized schnorr proofs. In A. Joux, editor, *Advances in Cryptology — EUROCRYPT 2009*, Lecture Notes in Computer Science . Springer Verlag, 2009.

[12] J. Camenisch and A. Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In M. K. Franklin, editor, *Advances in Cryptology — CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science* , pages 56–72. Springer Verlag, 2004.

[13] J. Camenisch and V. Shoup. Practical verifiable encryption and decryption of discrete logarithms. In D. Boneh, editor, *Advances in Cryptology — CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science* , pages 126–144, 2003.

[14] J. Camenisch and M. Stadler. Efficient group signature schemes for large groups. In B. Kaliski, editor, *Advances in Cryptology — CRYPTO '97*, volume 1296 of *Lecture Notes in Computer Science* , pages 410–424. Springer Verlag, 1997.

[15] I. Damgård and E. Fujisaki. An integer commitment scheme based on groups with hidden order. http://eprint.iacr.org/2001, 2001.

[16] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. In *Proceedings of the 13th USENIX Security Symposium*, 2004.

[17] A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In A. M. Odlyzko, editor, *Advances in Cryptology — CRYPTO '86*, volume 263 of *Lecture Notes in Computer Science* , pages 186–194. Springer Verlag, 1987.

[18] R. T. Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California, Irvine, 2000.

[19] J. Giles. Internet encyclopaedias go head to head. *Nature*, 438, 14 Dec. 2005.

[20] IBM. Cryptographic protocols of the Identity Mixer library, v. 1.0. IBM Research Report RZ3730, IBM Research, 2009. http://domino.research.ibm.com/library/cyberdig.nsf/index.html.

[21] C. Lampe and P. Resnick. Slash(dot) and burn: Distributed moderation in a large online conversation space. In *ACM CHI 2004 Conference on Human Factors in Computing Systems*, 2004.

[22] E. Pavlov, J. Rosenschein, and Zvi. Supporting privacy in decentralized additive reputation systems. In *iTrust 2004*, 2004.

[23] T. P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In J. Feigenbaum, editor, *Advances in Cryptology – CRYPTO '91*, volume 576 of *Lecture Notes in Computer Science* , pages 129–140. Springer Verlag, 1992.

[24] S. Steinbrecher. Design options for privacy-respecting reputation systems within centralised internet communities. In *SEC*, pages 123–134, 2006.

[25] F. B. Viégas, M. Wattenberg, and K. Dave. Studying cooperation and conflict between authors with history flow visualizations. In *Proceedings of the 2004 Conference on Human Factors in Computing Systems, CHI 2004*, 2004.

[26] L. von Ahn. Games with a Purpose. *IEEE Computer Magazine*, June 2006.

A CL-Signature E-Cash

We can easily implement a simple version of e-cash by using anonymous credentials.

- $(sk_{\mathsf{B}}, pk_{\mathsf{B}}) \leftarrow \mathsf{SetupBank}(\ell)$
- $(\sigma_\Psi, d_\Psi, s_\Psi)() \leftarrow \mathsf{Withdraw}(\sigma_U; sk_{\mathsf{U}}, pk_{\mathsf{B}})(sk_{\mathsf{B}}, pk_{\mathsf{I}})$
  - $s_\Psi, d_\Psi, r \xleftarrow{\$} \mathbb{Z}_q$;
  - $D \leftarrow \mathsf{Commit}((sk_{\mathsf{U}}, s_\Psi, d_\Psi), r)$;
  - $SPK\{(\sigma_U, sk_{\mathsf{U}}, s_\Psi, d_\Psi, r) :$
    $\mathsf{VerifyCommit}(D, (sk_{\mathsf{U}}, s_\Psi, d_\Psi), r) \wedge$
    $\mathsf{VerifySign}(\sigma_U, sk_{\mathsf{U}}; pk_{\mathsf{I}})$
    $\}$;
  - $(\sigma_\Psi)() \leftarrow \mathsf{HiddenSign}((D), (), r; pk_{\mathsf{B}})(sk_{\mathsf{B}})$;
  - Output: $(\sigma_\Psi, d_\Psi, s_\Psi)()$
- $(T, R)(\Psi) \leftarrow \mathsf{Spend}(\sigma_U, (\sigma_\Psi, d_\Psi, s_\Psi); sk_{\mathsf{U}})(pk_{\mathsf{B}})$
  - $R \xleftarrow{\$} \mathbb{Z}_q$;
  - $T \leftarrow g_B^{sk_{\mathsf{U}}\mathrm{R}} g_B^{d_\Psi}$;
  - $SPK\{(\sigma_U, sk_{\mathsf{U}}, \sigma_\Psi, s_\Psi, d_\Psi, R) :$
    $\mathsf{VerifySign}(\sigma_U, (sk_{\mathsf{U}}); pk_{\mathsf{I}}) \wedge$
    $\mathsf{VerifySign}(\sigma_\Psi, (sk_{\mathsf{U}}, s_\Psi, d_\Psi); pk_{\mathsf{B}}) \wedge$
    $T = g_B^{sk_{\mathsf{U}}\mathrm{R}} g_B^{d_\Psi}$
    $\}$;
  - $\Psi \leftarrow (s_\Psi, R, T, \Phi)$
  - Output: $(T, R)(\Psi)$
- $()(\Psi) \leftarrow \mathsf{Deposit}(\Psi)()$
- $(pk_{\mathsf{U}}, \Pi) \leftarrow \mathsf{Identify}(\Psi_1, \Psi_2)$
- 0 or $1 \leftarrow \mathsf{VerifyGuilt}(pk_{\mathsf{U}}, \Pi)$

B EC Protocols based on the ℓ-SDH Assumption

We provide the details of a concrete instantiation of the incentive system in the signature scheme by Au et al. [3]. It is secure under the ℓ-SDH Assumption [6], which states that in a group $G = \langle g \rangle$ of order p it is hard to compute $(c, g^{1/(x+c)})$ given $g, g^x, \ldots, g^{x^\ell}$. The scheme is based on Camenisch and Lysyankaya [12] and of Boneh et al. [7].

It assumes cyclic groups $\mathbb{G}$ and $\mathbb{G}_\mathrm{T}$ of order p and a bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_\mathrm{T}$. The signer's secret key is a random element $x \xleftarrow{\$} \mathbb{Z}_q$. The public key contains a number of random bases $g_1, h_0, \ldots, h_\ell, h_{\ell+1} \xleftarrow{\$} \mathbb{G}$, where $\ell \in \mathbb{N}$ is a parameter, and $y := g_1^x$.

A signature on messages $m_0, \ldots, m_\ell \in \mathbb{Z}_p$ is a tuple (A, r, s) where $r, s \xleftarrow{\$} \mathbb{Z}_p$ are values chosen at random by the signer and $A := (g_1 h_0^{m_0} \cdots h_\ell^{m_\ell} h_{\ell+1}^r)^{1/(x+s)}$. Such a signature can be verified by checking whether $e(A, g_1^s y) = e(g_1 h_0^{m_0} \cdots h_\ell^{m_\ell} h_{\ell+1}^r, g_1)$.

Now assume that we are given a signature (A, r, s) on messages $m_0 \ldots, m_\ell \in \mathbb{Z}_p$ and want to prove that we indeed possess such a signature. To this end, we need to augment the public key with values $u, v \in \mathbb{G}$ such that $\log_{g_1} u$ and $\log_{g_1} v$ are not known. This can be done as follows.

1. Choose random values $t, t' \xleftarrow{\$} \mathbb{Z}_p$ and compute $\tilde{A} := Au^t$, $B := v^t u^{t'}$.
2. Execute the following proof of knowledge (where $\alpha = st$ and $\beta = st'$)

$$PK\{(\alpha, \beta, s, t, t', m_0, \ldots, m_\ell, r) : \quad B = v^t u^{t'} \ \wedge \ 1 = B^{-s} v^\alpha u^\beta \ \wedge \ \frac{e(\tilde{A}, y)}{e(g_1, g_1)} = e(u, y)^t e((\tilde{A}^{-s} u^\alpha h_{\ell+1}^r \prod_{i=0}^{\ell} h_i^{m_i}), g_1)\} \ .$$

It was proved in [3] that the above signature is unforgeable under adaptively chosen message attack if ℓ-SDH assumption holds, where ℓ is the number of signature queries.

B.1 Setup

We provide three setup functions for the privacy-friendly incentive system: (i) users and banks generate their respective keys. (ii) Each user U obtains an anonymous credential, that is a Camenisch-Lysyanskaya signature on her secret key sk_U. (iii) And, we establish a domain pseudonym system with articles and context-specific URLs as domains.

Firstly, let us consider the key generation. The algorithm BKEYGEN creates BBS CL signature system parameters, the bank's secret key x and public key $(g_1, h_0, \ldots, h_\ell, h_{\ell+1}, u, v, y)$ according to Section 2. The algorithm UKEYGEN generates a user's master secret key sk_U and the corresponding public key $pk_\mathsf{U} := g_1^{sk_\mathsf{U}}$. Secondly, we consider the identity certification. The user's secret key sk_U also serves as unique identity. It is certified in a Camenisch-Lysyanskaya signature $\sigma_U = (A, r, c)$ together with the user's roles and attributes. As a convention, we dedicate the zeroth signed message to the master key sk_U and subsequent messages m_1, …, m_ℓ to attributes and roles. Thirdly, we establish a domain pseudonym for a document/user combination upon a user interaction with that document. To that end, we assume existence of a cryptographic hash function $\mathcal{H}(\cdot)$ with domain $\mathbb{G}$: $\mathcal{H} : \{0,1\}^* \to \mathbb{G}$. We compute a domain pseudonym $N_{P,U}$ on a page P for a user U as $N_{P,U} := (\mathcal{H}(P))^{sk_\mathsf{U}}$. Given that $\mathcal{H}(P)$ is a random generator for $\mathbb{G}$, $(\mathcal{H}(P))^{sk_\mathsf{U}}$ will be a non-degenerate random group element with with overwhelming probability. Authors and editors need to submit their domain pseudonym for each transaction and proof possession of the corresponding secret key.

With this setup, we construct the e-coins of our system—(i) unspent ones $(\sigma_\Psi, s_\Psi, d_\psi)$ and (ii) spent ones $\Psi = (s_\Psi, R, T, \Phi)$—as follows: Firstly, an unspent e-coin is a Camenisch-Lysyanskaya signature on the coin's serial number s_Ψ and the owner's identity sk_U. The owner stores the serial number s_Ψ and his double-spending randomness d_Ψ in addition to the signature. Secondly, the recipient of a spent e-coin stores the tuple of serial number s_Ψ, SPEND protocol challenge R, double-spending value T, and proof transcript Φ.

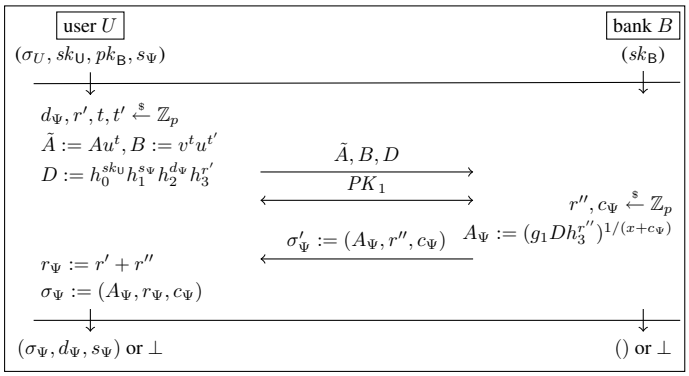


Figure 4: WITHDRAW protocol.

B.2 Withdraw

We use a coin WITHDRAW protocol in two instances: Firstly, a user of an Wiki document withdraws e-coins from a bank to offer incentives to editors. Secondly, the Wiki withdraws new coins in exchange to offered coins. We require in both cases that these e-coins are bound to the new owner's identity in order to ensure double-spending prevention. To ensure that, we proceed as follows and depict the whole flow in Figure 4.

The user's credential $\sigma_U = (A, r, c)$ is a signature on the user's master key sk_U and and roles the user can have in the system. The user constructs a commitment on coin parameters and her identity sk_U, and then proves knowledge of its representation and identity. Figure 4 defines this protocol: the user computes the blinding $(\tilde{A}, B)$ of her identity credential σ_U and a commitment D on the coin parameters sk_U, s_Ψ, and d_Ψ.

Then, the user proves the possession of credential σ_U, the structure of D, and the equality of sk_U between σ_U and D, which Figure 4 references as PK_1:

$$
\begin{aligned}
PK_1\{(\alpha, \beta, \gamma, \tau, \tau', \mu_0, (\mu_i)_{1\le i\le \ell}, \varsigma_\Psi, \rho, \delta_\Psi, \rho') :& \\
D \equiv h_0^{\mu_0} h_1^{\varsigma_\Psi} h_2^{\delta_\Psi} h_3^{\rho'} \pmod p \wedge\ & B = v^\tau u^{\tau'} \ \wedge\ 1 = B^{-\gamma} v^\alpha u^\beta \ \wedge \\
\frac{e(\tilde{A}, y)}{e(g_1, g_1)} = e(u, y)^\tau e((\tilde{A}^{-\gamma} u^\alpha h_{\ell+1}^\rho h_0^{\mu_0} & \prod_{i=1}^{\ell} h_i^{\mu_i}), g_1)\}
\end{aligned}
$$

The bank verifies the correctness of the proof of knowledge and computes a preliminary BBS Camenisch-Lysyanskaya signature $\sigma'_\Psi := (A_\Psi, r'', c_\Psi)$ on the commitment D. The user, in turn, derives the coin by completing the signature σ_Ψ and storing the parameters s_Ψ and d_Ψ.

B.3 Submit Offer and Review

A user of the Wiki can offer incentive e-coins to editors by executing the SUBMIT OFFER operation. The Wiki rewards an editor with offered coins in the SUBMIT REVIEW operation. Both operations translate to spending the e-coins with the Wiki as clearing house.

B.3.1 Entitlements and separation of duty.

We compose the proof statements for the entitlement and separation of duty properties with the e-coin spending protocol below. They are highly specific to the Wiki's entitlement and SoD policies. Thus, we name general techniques.

We realize role-based and attribute-based entitlement by certifying the roles and attributes in the user's identity credentials or separate role credentials on the same master secret key. We prove the entitlement by selective disclosure of the attributes in the proof of knowledge.

We realize separation of duty with the document-centric domain pseudonym. Recall that the domain

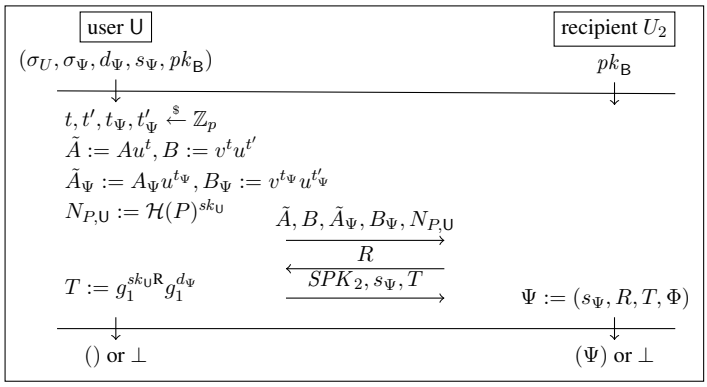


Figure 5: SUBMIT OFFER protocol.

pseudonym is on the user's master key: $g_{\mathcal{H}(P)}^{sk_\mathsf{U}}$. This does not only allow us a per-document linking[12], but also to prove inequality of entities in SUBMIT REVIEW. To prove a separation of duty between author U and editor R, the editor proves possession of her credential and representation of her domain pseudonym $N_{P,\mathsf{R}} = g_{\mathcal{H}(P)}^{sk_\mathsf{R}}$. Thereby, the Wiki can be satisfied that the domain pseudonym $N_{P,\mathsf{R}}$ for the reviewed document P is linked to the editor R and that it is unequal from the domain pseudonym of the author $N_{P,\mathsf{U}}$.

B.3.2 Spending an incentive e-coin.

The actual spending of an incentive e-coin is relatively straight-forward. Let us assume that a user holds a unspent coin Ψ' consisting of a serial number s_Ψ, a double spending randomness d_Ψ, and a coin signature $\sigma_\Psi = (A_\Psi, r_\Psi, c_\Psi)$. In order to spend such a coin with another user, the user needs to prove the possession of the coin signature σ_Ψ and establish a double spending equation $T = g_1^{sk_\mathsf{U}\mathsf{R}} g_1^{d_\Psi}$. We proceed with the outline of the SUBMIT OFFER protocol and present its definition in Figure 5.

Note that in SUBMIT REVIEW, the Wiki spends the incentive e-coin with the editor, yet the editor is proving the fulfillment of entitlement and separation of duty policy back to the Wiki before receiving the coin. We leave the detailed specification of this protocol to the extended version of this paper.

The user computes a blinding on her identity credential $\sigma_U = (A, r, c)$ and coin signature $\sigma_\Psi = (A_\Psi, r_\Psi, c_\Psi)$. Upon receiving a challenge R from the recipient, the user computes the double spending equation $T := g_B^{sk_\mathsf{U}\mathsf{R}} g_B^{d_\Psi}$. The user proves the possession of the identity credential σ_U and of the coin signature σ_Ψ as well as the structure of the double spend equation for T and the equality of the master key sk_U:

$$
\begin{aligned}
SPK_2\{&(\alpha, \alpha_\Psi, \gamma, \tau, \tau', \beta, \beta_\Psi, \gamma_\Psi, \tau_\Psi, \tau'_\Psi, \mu_0, (\mu_i)_{1\le i\le \ell}, \varsigma_\Psi, \rho, \delta_\Psi, \rho_\Psi) : \\
&B = v^\tau u^{\tau'} \ \wedge\ 1 = B^{-\gamma} v^\alpha u^\beta \ \wedge\ B_\Psi = v^{\tau_\Psi} u^{\tau'_\Psi} \ \wedge\ 1 = B_\Psi^{-\gamma_\Psi} v^{\alpha_\Psi} u^{\beta_\Psi} \ \wedge \\
&\frac{e(\tilde{A}, y)}{e(g_1, g_1)} = e(u, y)^\tau e((\tilde{A}^{-\gamma} u^\alpha h_{\ell+1}^\rho h_0^{\mu_0} \prod_{i=1}^{\ell} h_i^{\mu_i}), g_1) \wedge \\
&\frac{e(\tilde{A}_\Psi, y)}{e(g_1, g_1)} = e(u, y)^{\tau_\Psi} e((\tilde{A}_\Psi^{-\gamma_\Psi} u^{\alpha_\Psi} h_3^{\rho_\Psi} h_0^{\mu_0} h_1^{\varsigma_\Psi} h_2^{\delta_\Psi}), g_1) \wedge \\
&N_{P,\mathsf{U}} = \mathcal{H}(P)^{\mu_0} \wedge \\
&T \equiv (g_1^R)^{\mu_0} g_1^{\delta_\Psi} \pmod p; \\
&\}(\text{context}, R)
\end{aligned}
$$

This first three lines of terms in this proof show that the user possesses an identity signature and an e-coin signatures, both being bound to the same user's secret identity sk_U. The proof of representation of the domain pseudonym $N_{P,\mathsf{U}}$ and equality of the user's master secret key provide the separation of duty capabilities as described above. The last line proves the correctness of the double-spending equation, and in particular, that

[12] e.g., to allow the author to submit an offer for her document under special conditions.

the double-spending value T is also bound to the user's secret identity sk_{U} and computed correctly from R and the appropriate values contained in the e-coin signature.

The recipient concludes the protocol by storing the coin (s_Ψ, R, T, Φ), where Φ is the transcript of the signature proof of knowledge SPK_2. The remaining elements have the following meaning: s_Ψ is the coin's serial number, R the challenge randomness used in the SUBMIT OFFER operation, and T the double spending value.

B.4 Deposit

In order to DEPOSIT a coin, the recipient sends the coin $\Psi = (s_\Psi, R, T, \Phi)$ to the bank. The bank checks that it does not hold the serial number s_Ψ, yet. Otherwise, it engages in the double spending detection protocol IDENTIFY.

We opt-in for the explicit exchange of e-coins to support *Anonymity* and *Unlinkability*: the Wiki exchanges the e-coin for real money and WITHDRAW an e-coin independently from the DEPOSIT operation.

B.4.1 Double-spending Detection: Identify and VerifyGuilt.

We propose a public-key recovery mechanism to enable disciplining double-spending and ensuring the *Incentive Security/Balance* property from Section 4.2. As part of the DEPOSIT protocol, a bank can detect that a single coin was spent twice, because it receives two deposits with the same serial number s_Ψ. It can follow-up with disciplining perpetrators by executing the IDENTIFY algorithm and reconstructing two double-spending equations T_1 and T_2 from its payment records and solving this system of two equations with two unknown variables pk_U and b for the result pk_U. IDENTIFY takes two coins $\Psi_1 = (s_\Psi, R_1, T_1, \Phi_1)$ and $\Psi_2 = (s_\Psi, R_2, T_2, \Phi_2)$ as input. It outputs pk_{U} of the double-spender and the proof Π, if both coins hold the same serial number s_Ψ, $\perp$ otherwise. IDENTIFY derives the double-spender's public key as $\mathrm{pk}_U := (T_1/T_2)^{(R_1-R_2)^{-1}}$.

IDENTIFY algorithm constructs proof $\Pi = (\Psi_1, \Psi_2)$. From these values, any participant can publicly verify that the user with pk_{U} is indeed guilty of double-spending. Verifiers can recompute the equation above and check that the proof transcripts Φ_1 and Φ_2 are valid. We call this operation VERIFYGUILT.

C Protocols based on the Strong RSA Assumption

C.1 Initialization

1. Generate public key pk_U and private key sk_U (master secret) for the User. Section 4.5 states that the master secret should be chosen uniformly at random from the interval $[1, \rho]$. User public key $pk_U = g^{sk_U}$.
2. Generate public key and private key for the Bank.
3. Generate public key and private key for the Wikipedia.
4. Prove correctness of the public key of the issuer to the clients (users).

C.1.1 Group Parameters

The class `GroupParameters` must be made public to all parties. User must verify that ρ and Γ are prime and that $\rho \mid (\Gamma - 1)$, $\rho \nmid \frac{\Gamma-1}{\rho}$. User must retrieve the instance of the `GroupParameters` class from the server (issuer) and check the above properties. User must retrieve the instance of the `SystemParameters` class from the server (issuer) and check the constraints.

C.1.2 Issuer key generation

Issuer generates its public and private key. It needs to prove to a client (user), that the public key was generated correctly.

$$\begin{aligned} & SPK\{(x_Z, x_{R_1}, \ldots, x_{R_l})\}: \\ & \quad Z = S^{x_Z}, R_1 = S^{x_{R_1}}, \ldots, R_l = S^{x_{R_l}} \\ & \} \end{aligned} \tag{1}$$

C.1.3 User certificate issuance

Client (user) needs also issuer's public key for the certificate to be issued. After the certificate is issued, issuer's public key is part of the certificate.

C.2 Withdraw

C.2.1 User

1. Prove possession of the identity credential $\sigma_U = (A, e, r)$, signature on the user's master key sk_U and roles (additional attributes).
2. Construct a commitment on coin parameters s_Ψ, d_Ψ and user's identity sk_U.
3. Prove structure of the commitment C and the equality of sk_U.
4. Compute domain pseudonym $N_{D,U} = \mathcal{H}(D)^{sk_\mathsf{U}}$ using bank url as domain D.

$$
\begin{aligned}
&SPK_1\{(e, v, r', sk_{\mathsf{U}}, s_\Psi, d_\Psi, \{m_i | i \in I_c \cup I_h\}) : \\
&\quad Z\tilde{A}^{-2^{l_e-1}} \prod_{i \notin I_c \cup I_h} R_i^{-m_i} \equiv \pm \tilde{A}^e S^v \prod_{i \in I_c \cup I_h} R_i^{m_i} \pmod{n} \wedge && (2)\\
&\quad C \equiv S^{r'} R_0^{sk_{\mathsf{U}}} R_1^{s_\Psi} R_2^{d_\Psi} \pmod{n} \wedge && (3)\\
&\quad N_{D,U} = \mathcal{H}(D)^{sk_{\mathsf{U}}} \pmod{\Gamma} \wedge && (4)\\
&\quad sk_{\mathsf{U}}, s_\Psi, d_\Psi \in \{0,1\}^{l_m + l_\phi + l_{\mathcal{H}} + 2} \wedge && (5)\\
&\quad m_i \in \{0,1\}^{l_m + l_\phi + l_{\mathcal{H}} + 2} \; (i \in I_c \cup I_h) \wedge && (6)\\
&\quad e \in \pm\{0,1\}^{l'_e + l_\phi + l_{\mathcal{H}} + 1} && (7)\\
&\}
\end{aligned}
$$

C.2.2 Bank

1. Verify the correctness of the signature on user's identity credential σ_U.
2. Compute preliminary Camenisch-Lysyanskaya signature $\sigma'_\Psi := (A_\Psi, e_\Psi, r'')$ on the commitment C.

C.2.3 User

1. Complete the signature $\sigma_\Psi := (A_\Psi, e_\Psi, r := r' + r'')$.
2. Store parameters s_Ψ and d_Ψ and the coin signature $sigma_\Psi$.

C.3 Spend

C.3.1 User

1. Prove possession of a coin signature $\sigma_\Psi = (A_\Psi, e_\Psi, r_\Psi)$ and identity credential $\sigma_U = (A, e, v)$.
2. Receive challenge R from the recipient.
3. Establish double spending equation $T = g_1^{sk_{\mathsf{U}} R + d_\Psi} \pmod{\Gamma}$.
4. Prove the structure of the double spending equation for T and the equality of the master key sk_{U}.
5. Compute domain pseudonym $N_{D,U} = \mathcal{H}(D)^{sk_{\mathsf{U}}}$ using recipient url as domain D.

$$
\begin{aligned}
&SPK_2\{(e, v, sk_{\mathsf{U}}, s_\Psi, d_\Psi, \{m_i | i \in I_c \cup I_h\}) : \\
&\quad Z\tilde{A}^{-2^{l_e-1}} \prod_{i \notin I_c \cup I_h} R_i^{-m_i} \equiv \tilde{A}^e S^v \prod_{i \in I_c \cup I_h} R_i^{m_i} \pmod{n} \wedge && (8)\\
&\quad Z\tilde{B}^{-2^{l_e-1}} \equiv \tilde{B}^e S^v R_0^{sk_{\mathsf{U}}} R_1^{s_\Psi} R_2^{d_\Psi} \pmod{n} \wedge && (9)\\
&\quad T \equiv (g_1^R)^{sk_{\mathsf{U}}} g_1^{d_\Psi} \pmod{\Gamma} \wedge && (10)\\
&\quad N_{D,U} = \mathcal{H}(D)^{sk_{\mathsf{U}}} \pmod{\Gamma} \wedge && (11)\\
&\quad sk_{\mathsf{U}}, s_\Psi, d_\Psi \in \{0,1\}^{l_m + l_\phi + l_{\mathcal{H}} + 2} \wedge && (12)\\
&\quad m_i \in \{0,1\}^{l_m + l_\phi + l_{\mathcal{H}} + 2} \; (i \in I_c \cup I_h) \wedge && (13)\\
&\quad e \in \pm\{0,1\}^{l'_e + l_\phi + l_{\mathcal{H}} + 1} && (14)\\
&\}
\end{aligned}
$$

C.3.2 Recipient

1. Verify correctness of the coin signature σ_Ψ and signature on user's identity credential σ_U.
2. Store coin $\Psi = (s_\Psi, R, T, \Phi)$.
   - s_Ψ - coin serial number
   - R - challenge randomness
   - T - double spending equation
   - Φ - transcript of the signature proof of knowledge SPK_2.

C.4 SubmitReview

C.4.1 User

1. Compute domain pseudonym $N_{P,U_2} = \mathcal{H}(P)^{sk_{\mathsf{U}_2}}$ for the page P.
2. Prove possession of the identity credential $\sigma_{U_2} = (A_2, e_2, v_2)$ and structure of the domain pseudonym N_{P,U_2}.

$$
\begin{aligned}
SPK_3\{&(e_2, v_2, sk_{\mathsf{U}_2}, \{m_i | i \in I_c \cup I_h\}) : \\
&Z\tilde{A}^{-2^{l_e}-1} \prod_{i \notin I_c \cup I_h} R_i^{-m_i} \equiv \tilde{A}^{e_2} S^{v_2} \prod_{i \in I_c \cup I_h} R_i^{m_i} \pmod{n} \wedge && (15)\\
&N_{P,U_2} = \mathcal{H}(P)^{sk_{\mathsf{U}_2}} \pmod{\Gamma} \wedge && (16)\\
&sk_{\mathsf{U}_2} \in \{0,1\}^{l_m + l_\phi + l_\mathcal{H} + 2} \wedge && (17)\\
&m_i \in \{0,1\}^{l_m + l_\phi + l_\mathcal{H} + 2} \; (i \in I_c \cup I_h) \wedge && (18)\\
&e_2 \in \pm\{0,1\}^{l'_e + l_\phi + l_\mathcal{H} + 1} && (19)\\
\}&
\end{aligned}
$$

C.4.2 Recipient

1. Verify correctness of the proof of knowledge.
2. Spend coins that were offered for a review.

C.4.3 User

1. Receive coins and verify their correctness.
2. Deposit coins at the bank and withdraw fresh ones.

C.5 Deposit

C.5.1 User

1. Compute domain pseudonym $N_{D,U_2} = \mathcal{H}(D)^{sk_{\mathsf{U}_2}}$ using bank url as domain D.
2. Prove possession of the identity credential $\sigma_{U_2} = (A_2, e_2, v_2)$ and structure of the domain pseudonym N_{D,U_2}.

3. Send the coin $\Psi = (s_\Psi, R, T, \Phi)$ to the bank.

$$
\begin{aligned}
&SPK_4\{(e, v, sk_{\mathsf{U}_2}, s_\Psi, d_\Psi, \{m_i | i \in I_c \cup I_h\}) : \\
&\quad Z\tilde{A}^{-2^{l_e}-1} \prod_{i \notin I_c \cup I_h} R_i^{-m_i} \equiv \tilde{A}^e S^v \prod_{i \in I_c \cup I_h} R_i^{m_i} \pmod{n} \wedge && (20)\\
&\quad N_{D,U_2} = \mathcal{H}(D)^{sk_{\mathsf{U}_2}} \pmod{\Gamma} \wedge && (21)\\
&\quad sk_{\mathsf{U}}, s_\Psi, d_\Psi \in \{0,1\}^{l_m + l_\phi + l_{\mathcal{H}} + 2} \wedge && (22)\\
&\quad m_i \in \{0,1\}^{l_m + l_\phi + l_{\mathcal{H}} + 2} \; (i \in I_c \cup I_h) \wedge && (23)\\
&\quad e \in \pm\{0,1\}^{l'_e + l_\phi + l_{\mathcal{H}} + 1} && (24)\\
&\}
\end{aligned}
$$

C.5.2 Bank

1. Verify the identity credential σ_{U_2} and the structure of the domain pseudonym N_{D,U_2}.
2. Check that there is no serial number s_Ψ already and update user's balance, store $s_P si$.
3. If there is serial number s_Ψ, engage in the double spending detection protocol (Identify).

C.5.3 Identify

1. Take two coins $\Psi_1 = (s_\Psi, R_1, T_1, \Phi_1)$ and $\Psi_2 = (s_\Psi, R_2, T_2, \Phi_2)$ as input.
2. Output the pk_{U} of the double-spender and the proof $\Pi = (\Psi_1, \Psi_2)$, if both coins hold the same serial number s_Ψ, $\perp$ otherwise.

$$
\begin{aligned}
T_1 &= \mathrm{pk}_U^{R_1} \cdot g_1^{b_\Psi} && (25)\\
T_2 &= \mathrm{pk}_U^{R_2} \cdot g_1^{b_\Psi} && (26)\\
(T_1/T_2)^{(R_1-R_2)^{-1}} &= \\
\left(\mathrm{pk}_U^{R_1} \cdot g_1^{b_\Psi} / \mathrm{pk}_U^{R_2} \cdot g_1^{b_\Psi}\right)^{(R_1-R_2)^{-1}} &= \\
\left(\mathrm{pk}_U^{R_1-R_2}\right)^{(R_1-R_2)^{-1}} &= \mathrm{pk}_U && (27)
\end{aligned}
$$

C.5.4 VerifyGuilt

Verifiers can recompute the equation above and check that the proof transcripts Ψ_1 and Ψ_2 are valid.

D Zero-Knowledge Proof of Knowledge Parameters

Parameter	PK Secret	Description
Proof of Knowledge of σ_U		
t, t'	τ, τ'	Blinding for PK of an identity signature σ_U
$A, \tilde{A}, B$		σ_U
r, r', r''	ρ, ρ', ρ''	Blinding for σ_U
c	γ	Signature-specific secret for σ_U
	α, β	Completion for PK σ_U
sk_{U}	μ_0	User identity/master key in σ_U
m_i	μ_i	Other attributes in σ_U, e.g., roles
Proof of Knowledge of a coin σ_Ψ		
t_Ψ, t'_Ψ	τ_Ψ, τ'_Ψ	Blinding for PK of σ_Ψ
$A_\Psi, \tilde{A_\Psi}, B_\Psi$		σ_Ψ
r_Ψ	ρ_Ψ	Blinding for coin signature σ_Ψ
c_Ψ	γ_Ψ	Signature-specific secret for σ_Ψ
	α_Ψ, β_Ψ	Completion for PK Ψ
s_Ψ	ς_Ψ	Coin serial number in Ψ
d_Ψ	δ_Ψ	Double spent randomness for Ψ
T		Double spent value for Ψ

E System Configuration Parameters

	parameters
U	id_{U}, pw_{U}, sk_{U}, pk_{U}; pk_{W}, pk_{B_0}, addr_{B_0}
W	sk_{W}, pk_{W}; pk_{B_0}, addr_{B_0}
R	id_{R}, pw_{R}, sk_{R}, pk_{R}; pk_{W}, pk_{B_0}, addr_{B_0}
B_0	id_{B_0}, pw_{B_0}, sk_{B_0}, pk_{B_0}; $\{\,\mathrm{pk}_{U_i}\,\}_i$