Journal of Information Processing Vol. 19 1–15 (July 2011)

Regular Paper

Sybil-Free Pseudonyms, Privacy and Trust: Identity Management in the Internet of Services

Leonardo A. Martucci,[†1,†2] Sebastian Ries[†1,†2]
and Max Mühlhäser[†1,†2]

We propose an identity management system that supports role-based pseudonyms that are bound to a given set of services (service contexts) and support the use of reputation systems. Our proposal offers a solution for the problem of providing privacy protection and reputation mechanisms concurrently. The trust information used to evaluate the reputation of users is dynamic and associated to their pseudonyms. In particular, our solution does not require the support or assistance from central authorities during the operation phase. Moreover, the presented scheme provides inherent detection and mitigation of Sybil attacks. Finally, we present an attacker model and evaluate the security and privacy properties and robustness of our solution.

1. Introduction

The "Internet of the Services" depicts the Internet as a conglomerate of interconnected services that interact and cooperate to fulfill tasks provided by users. The Internet of Services must be built on both security, privacy and trust establishment.

In service-oriented environments everyone is allowed to offer services, and there will be numerous competing service providers offering services of a similar nature. Whenever a customer has the choice between two or more services, e.g., e-books, music, or video, quality of a service is a key factor. The concepts of trust and reputation have been shown to be promising concepts to support customers in such situations in selecting a high quality service 24),25),35), as they help to assess services based on past experience gained by the user community.

Service-oriented networks also need to offer security and privacy guarantees to their users. In particular, personal information must be processed and transmitted according to the applicable privacy-protection legislation or regulation, such as the European Directive 95/46/EC 1) in Europe. Privacy is best protected in the absence of personal identifiable information, i.e., anonymity. Moreover, users' privacy protection requires unlikable actions, i.e., an observer should not be able to link two or more actions to the same user. However, building up trust and reputation usually requires long-term identifiers which can be linked over numerous transactions. At a first glance, this seems to be in conflict with the protection of the users' privacy, as anonymity and unlinkability are key properties when referring to privacy protection.

The design of an identity management scheme that addresses the conflicting requirements of privacy and reputation is the main contribution of this paper. Our solution is built following the second law of identity 13),14) where the minimal amount of personal information is disclosed and its use in our identity management scheme is limited.

We propose a system architecture for generating role-based pseudonyms that are bound to a given set of services, called a service context. Those pseudonyms provide the means for decoupling real world identities from the digital identifiers that are used as a basis for the trust establishment. Our proposal is independent from central authorities or trusted third parties during the operation phase and it provides the means for the efficient detection of *Sybil identifiers*. Naturally, there are proposals of fully distributed solutions based on the concept of a Web of Trust 45). However, such solutions provide only weak authentication 31) because of the initial assumption of trust transitiveness used for authentication purposes. Fully distributed solutions cannot be proved to be secure as they are prey to identity attacks and, thus, are not further discussed in this paper.

The paper is organized as follows: Section 2 presents the scenario and the objective of our approach. Section 3 introduces the basic building blocks. Section 4 presents the identity management scheme designed after the definition of the system requirements. The security evaluation is presented in Section 5. Section 6 outlines the efficient Sybil detection mechanism and Section 7 discusses the use of pseudonyms for services and also discusses the advantages and disadvantages

[†1] Technische Universität Darmstadt, Telecooperation Lab (TK)
[†2] Center for Advanced Security Research Darmstadt (CASED)

of other pseudonym constructions. Finally, Section 8 concludes the paper.

2. Application Scenario and Objectives

In the first part of this section we introduce the service-oriented scenario and the notation. Then, we define the system objectives regarding privacy and trust.

2.1 Internet of Services Scenario & Notation

In an Internet of Services scenario an arbitrary number of *service providers* $s_i \in \mathcal{S} \,|\, 1 < i < |\mathcal{S}|$ offer their services to a set of users $u_i \in \mathcal{U} \,|\, 1 < i < |\mathcal{U}|$. Furthermore, we introduce the concept of *service contexts* $\mathcal{C}^i \in \mathcal{C} \,|\, 1 < i < |\mathcal{C}|$, and we group all services with similar nature, e.g., sellers of books or online web space providers in one *service context*[⋆1]. Setting up service contexts is a natural consequence of an Internet of Services environment, where services competing for users are published, i.e., listed, in service directories. Within each service context $\mathcal{C}^i$, there are two sets of identifiers; the set $\mathcal{S}^i$ refers to the identifiers of the service providers available in this context, and the set $\mathcal{P}^i$ refers to the identifiers of the customers that want to use services in this context, where $|\mathcal{P}^i| \leq |\mathcal{U}|$. For customers in $\mathcal{U}$ we propose an identity management scheme, that allows each customer to create a unique pseudonym[⋆2] $p_u^{\mathcal{C}^i}$ per service context $\mathcal{C}^i$. The relationship between the sets $\mathcal{U}$, $\mathcal{S}$, $\mathcal{C}^i$, $\mathcal{P}^i$, and $\mathcal{S}^i$ is illustrated in **Fig. 1**. A summary of the notation is presented in **Table 1**.

2.2 Objectives

In general, we see two approaches to tackle the conflict of establishing trust while preserving privacy.

The first one uses pseudonyms in order to prevent histories from being linked to a user's real identity. Here, role-based pseudonyms provide a means for establishing trust between users within different service contexts $\mathcal{C}^1, \mathcal{C}^2, \ldots$ by introducing different, unlinkable pseudonyms per user per context. Thus, a user u_i can learn that the owner u_j of a certain pseudonym is active in a context $\mathcal{C}^k$ and whether u_j shows trustworthy or untrustworthy behavior in this context $\mathcal{C}^k$. However, as the pseudonyms are not linkable across contexts, user u_i cannot learn whether u_j is active in any other service context $\mathcal{C}^l$ (with $k \neq l$) nor about his behavior or preferences in this other context.

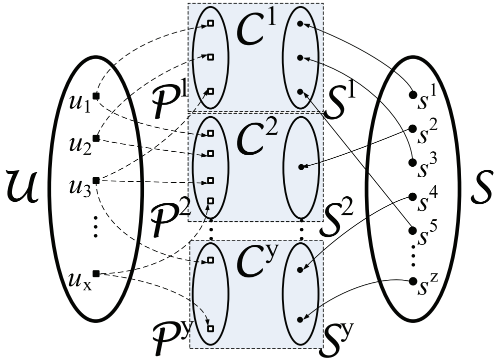

Fig. 1 Identifiers and service contexts.

Table 1 General notation.

s_i	Service provider i.
u_j	User (customer) j.
$\mathcal{S}$	Set of service providers s.
$\mathcal{U}$	Set of users (customers) u.
$\mathcal{C}^k$	Service context k.
$p_{u_j}^{\mathcal{C}^k}$	Pseudonym of user u_j for service context $\mathcal{C}^k$.
$\mathcal{S}^k$	Subset of $\mathcal{S}$ that only contains the services available for context $\mathcal{C}^k$.
$\mathcal{P}^k$	Set of pseudonyms that are used in context $\mathcal{C}^k$.

A second approach protects the user's privacy by preventing that the recipient of a set of recommendations can read the content of the separate recommendations by applying homomorphic encryption and random numbers for masking the recommendations. As current solutions following this approach[20)] prevent the recipient from learning whether individual recommenders tend to provide accurate recommendations or misleading ones, we follow the first approach in this paper.

The objective of our proposal is to offer a privacy-friendly identity management scheme with support to evidence-based trust or reputation systems for service environments. The requirements of such a scheme are threefold:

(i) Providing unique, long-term identifiers as a basis for a trust or reputation

[⋆1] Note that the parameters $|\mathcal{S}|$, $|\mathcal{U}|$, and $|\mathcal{C}|$ may change over time, however, in the paper we treat them as static parameters for the simplicity of the notation.

[⋆2] A pseudonym is an identifier of a subject other than one of the subject's real names[32)].

model. These identifiers are needed as a basis for building histories on the quality of used services and on the behavior of others users when providing recommendations about service providers within a certain service context.

(ii) Providing unlinkability between the users' behaviors in different service contexts. This requirement protects the users' privacy, in the sense that they can have different behaviors in different service contexts; but others cannot link a user's pseudonyms across service contexts (as long as the user does not create Sybil identifiers). An adversary is thus not able to track the users' actions in different contexts, and therefore is not able to link a user's partial identities[⋆1].

(iii) Being able to detect *Sybil identifiers*[17]. This is an important feature as Sybil attacks are a threat to privacy and to trust models. The basic idea of a Sybil attack is that a single entity (the attacker) creates an arbitrary high number of seemingly independent identities. This poses a threat to privacy, as it may allow attackers to reduce the entropy of the anonymity set as there are no guarantees regarding the 1:1 relationship between digital and physical identifiers[27),28]. For instance, a user that is part of an anonymity set with cardinality n, would in principle have an action associated to her with a probability $P = (1/n)$. However, it is not possible to guarantee that the other $(n-1)$ digital identifiers belong to $(n-1)$ different users. An adversary controlling these $(n-1)$ identifiers would increase P to $P = (1/2)$, for any observer that is able to verify that there are only two users in the anonymity set, or $P = 1$, for the Sybil attacker perspective. Furthermore, a Sybil attack allows adversaries to increase their influence on the trust system to provide misleading recommendations by seemingly independent entities. Finally, a Sybil attack also allows adversaries to erase bad history as they could use a newly created identifier whenever they want to. In the context of trust models the latter type is usually referred to as *whitewashing*.

2.3 Trust and Privacy Requirements and the Application Scenario

In an Internet of Services scenario, trust and privacy are both fundamental properties that need to be guaranteed. Trust models for building up reputation systems and privacy-enhancing technologies for protecting users' personal information from being abused or misused. The three aforementioned requirements (in Section 2.2) are essential for building up an identity management scheme that supports both trust and privacy.

The requirement regarding unique, long-term identifiers is a basic assumption of reputation schemes. They are needed to build up histories regarding both services and users of the system. Services and users should not be allowed to erase their history data just by deleting an identifier and creating a new one (i.e., whitewashing, also related to Sybil attacks). In the Internet of Services scenario, this problem arises on the calculus of the reputation of services from recommendations from other users. If no long-term identifiers are used, the weight of other users's recommendations cannot be calculated properly if there are no history data available regarding previous recommendations.

Unlinkability between the users' behaviors in different service contexts protect users' privacy by decoupling potential personal identifiable information traces from different service contexts and, thus, prevents the profiling of users. Reputation systems also benefit from users' unlinkability between different service contexts by associating a single partial identity to a given service context. Thus, apart from protecting users' privacy, this requirement allows a user to be a good recommender for a set of services $\mathcal{S}_i \in \mathcal{S}$ and a bad recommender for another set of services $\mathcal{S}_j \in \mathcal{S}$.

Detection and prevention of Sybil attacks is fundamental for both privacy protection and for reputation systems. As previously mentioned in Section 2.2, Sybil attacks reduce the entropy of anonymity sets. Hence, users cannot be assured of their privacy protection level if the cardinality of the anonymity set cannot be defined. Reputation schemes also need to be able to detect and prevent Sybil attacks, since malicious users could use Sybil attacks to arbitrarily improve or decrease the reputation of services just by creating a large number of identities. For example, imagine a scenario in which a service provider s_1 that competes with a service provider s_2 creates an arbitrary number of fake users (i.e., Sybil identifiers) for providing misleading recommendations about s_2.

[⋆1] Each partial identity represents a user in a specific context or role[32].

3. Basic Concepts and Background

In this section, we briefly introduce the basic concepts that we propose as a basis for the trust establishment and for the identity management scheme. First, we describe the concepts of trust and reputation

3.1 Trust

In the setting introduced above, trust and reputation models are important means for supporting users when selecting a service provider. For a definition of trust we refer to the definition of *reliability trust* in Ref. 25): "Trust is the subjective probability by which entity expects that another entity performs a given action on which its welfare depends." Evidence-based trust and reputation are similar concepts and in computational models both are usually based on the history of past interactions. In this paper, we focus on trust as a user's subjective expectation about a service provider, and not on reputation, which is considered to be a more objective value that would be shared by all entities in a community.

Going along with the definition of trust, we argue for the use of probabilistic trust models, as in those models the trust value has a clear semantics, and in addition it can be used in order to judge whether it is rational to interact at all – given the possible benefits and the possible costs – based on the expected utility of the interaction [23),33),35)]. Here, we like to especially refer to Bayesian trust models as they naturally allow for the interpretation of trust as a subjective probability, which allows for the consideration of personal preferences and context-dependent parameters (for details see e.g., Refs. 10), 24), 34), 37), 39), 44))[⋆1].

3.2 Building Blocks and Process of Trust Establishment

In the following, we present the building blocks of a (distributed) trust system and the basic idea for the process of trust establishment in such a system. We refer to the participants, i.e., users and service providers, in the system as *entities*. *Interactions* are actions between entities, i.e., the usage of a service or a capability that is offered by a service provider, e.g., buying goods or information. Thus, the type of interaction specifies the *service context*, in which a user wants to interact with a service provider.

Whenever, an entity A is in the role of the initiator of an interaction, i.e., entity A has to select a service provider from a set of available service providers, it may evaluate the trustworthiness of the available service providers as a basis for the selection. Hereby, entity A uses its direct evidence from previous interactions and recommendations (also called indirect evidence).

Having collected direct evidence and recommendations about one or multiple service providers, computational trust model provide a means for aggregating the evidence – hereby removing or giving lower weight to recommendations from unreliable sources – and deriving trust values for the service providers, which then can be the basis for the decision on whether to interact with one of the available service providers at all, and which service provider to select. After an interaction, the information on the quality of this interaction can be used to update one's direct evidence about the behavior of this service provider, and to update the trust in one's recommenders based on the accuracy of their recommendations.

3.3 Implications of this Process of Trust Establishment

Treating trust as a subjective value and calculating it in a distributed manner has a number of advantages, especially when considering privacy aspects:

First, it does not require a trusted third party or trusted distributed mechanism that collects and aggregates ratings. Especially, this means that when a user rates an interaction, this piece of information is at first and foremost a private one.

This leads directly to the second advantage: The private rating after an interaction can be used to decide whether the received recommendations had been accurate in relation to the subjective rating or not. As the user's private rating can be expected to be subjectively accurate, this disburdens distributed, subjective trust models from estimating whether a rating would be objectively accurate, which is usually necessary in centralized reputation systems, e.g., Ref. 44).

Third, users are free to decide to whether to distribute recommendations and to whom. This is important, as the provision of recommendations may be linked to information which the recommender can consider to be private. For example, when providing recommendations honestly, the recipient of a recommendation can learn about the recommender's previous interaction partners, the recommender's way of rating, and in the end about the recommender's personal pref-

[⋆1] In the most simple version, the aggregation of direct evidence and recommendations would lead to a number of positive evidence units r and a number of negative evidence units s, and the trust value of a service provider would be calculated as $(r+1)/(r+s+2)$.

erences.

3.4 Cryptographic Tools for Pseudonym Construction

Different cryptographic systems can be used to create unlinkable and unique pseudonyms (see requirement (ii) in Section 2.2). As long as the identification of "double-spent" pseudonyms is not an issue, such pseudonyms can be realized based on the so-called epoch number of direct anonymous attestation[9]. By binding a different tag to every identity domain, k-times anonymous authentication[40] can be used to create unique pseudonyms.

To achieve the objectives (ii) and (iii) presented in Section 2.2, we use a cryptographic construction for creating self-certified Sybil-Free pseudonyms[27],[30]. Self-certified Sybil-free pseudonyms are obtained by a non-interactive publicly verifiable variant of a special signature scheme originally introduced for periodic n-times spendable e-tokens[11]. In our approach we use $k = 1$, so that each user is represented by at most one pseudonym per service context. In addition, a freshly generated public key is bound to each pseudonym. Moreover, Sybil-free pseudonyms are produced through a mechanism of self-certification.

This mechanism uses different cryptographic building blocks and primitives, such as anonymous credentials and group signatures, for generating an arbitrary number of pseudonyms $p_i \in \mathcal{P}$, where $i \in \mathbb{N}$, from one initial identifier $u \in \mathcal{U}$, which is obtained from a trusted third party (TTP) in the bootstrapping phase. The generation of the self-certified Sybil-free pseudonyms also produces a certificate associated with the self-certified Sybil-free pseudonym that has the following uses[27],[30]:

- to bind a freshly generated public key to a pseudonym p_i. This operation is similar to the binding of public keys to X.509 certificates;
- to verify a pseudonym p_i and its binding to the aforementioned public key; and
- to disclose the initial identifier u, which is obtained from the TTP, and the revocation of the certificates obtained from it, if the user that owns it creates more than one pseudonym p for a given service context $\mathcal{C}^i$.

In **Table 2** we summarize the notation used in the cryptographic tools and algorithms presented in this and the following sections.

Table 2 Notation used for the pseudonym construction.

pk_{TTP}	The TTP's public key.
sk_{TTP}	The TTP's secret key.
pk_u	The user's public key.
sk_u	The user's secret key.
u	The initial identifier, which is realized as an e-token dispenser, and it is signed by the TTP.
r_u	Revocation information of u stored by the TTP.
$pk_{u,\mathcal{C}^i}$	Freshly generated public-key to be used in service context $\mathcal{C}^i$.
S	A pseudo-random pseudonym, which is a serial number.
τ	A pseudonym certificate, which is a transcript proving that S was properly generated from a valid identifier u.
p	The pseudonym, which is the triplet $(pk_{u,\mathcal{C}^i}, S, \tau)$.

3.5 Algorithms for Pseudonym Construction

The self-certified Sybil-free pseudonyms are an e-token based signature scheme that consists of the following eight algorithms[2],[27],[30]:

- *IKg* $(1^k) \rightarrow (pk_{TTP}, sk_{TTP})$ — this algorithm is used to create the issuer's, i.e., the TTP's, public and private key pair (pk_{TTP}, sk_{TTP}). The value k is the security parameter, where k is in unary, and 1^k denote the unary representation of integer k[21].
- *UKg* $(1^k, pk_{TTP}) \rightarrow (pk_u, sk_u)$ — this algorithm is used to create the user's public and private key pair (pk_u, sk_u).
- $(Obtain(pk_{TTP}, sk_u), Issue(pk_u, sk_{TTP})) \rightarrow (u, r_u)$ — the algorithms *Obtain* and *Issue* define a protocol between the users and the TTP. The algorithms are related and used to request and issue the initial identifier u. The algorithm *Obtain* is executed by users, while the algorithm *Issue* is executed by the TTP. At the end of this protocol, users obtain the initial identifier u, which is, basically, an e-token dispenser that can be used to create a pseudonym. The identifier u is an e-token dispenser comprised of a seed s for the pseudo-random function f_s, the secret key sk_u, and the Camenisch and Lysyanskaya (CL) signature[12] on (s, sk_u). CL signatures are used to prevent the TTP from learning about s or sk_u. The TTP stores the public key of the user (pk_u) and the revocation information r_u under the user's identity.
- *Sign* $(pk_{u,\mathcal{C}^i}, u, pk_{TTP}, \mathcal{C}^i) \rightarrow (S, \tau, u')$ — *Sign* outputs a pseudo-random pseudonym S (a token serial number), a pseudonym certificate τ (a transcript),

and an updated (e-token dispenser) u'. This algorithm is used to sign a freshly generated public key $pk_{u,\mathcal{C}^i}$. This freshly generated public key pair is used for authenticating user u and also for signing messages, i.e., recommendations. a user running the *Sign* algorithm uses the e-token dispenser to release a serial number $S = f_s(0\|\mathcal{C}^i)$, a double-show tag $E = pk_u \cdot f_s(1\|C^i)^{h(m)}$, and using the Fiat-Shamir heuristic[19] it creates a non-interactive zero-knowledge proof σ that (S, E) corresponds to a valid dispenser for the identity domain $\mathcal{C}^i$, i.e., the user proves in zero-knowledge that S and E were properly formed from values (s, sk_u) signed by the TTP. To sign $pk_{u,\mathcal{C}^i}$, it is hashed into the challenge together with the first message and the public parameters of the proof. The transcript τ contains both E and σ. An e-token is verified by checking the non-interactive proof. The triplet $(pk_{u,\mathcal{C}^i}, S, \tau)$ corresponds to a self-certified Sybil-free pseudonym p generated for a service context $\mathcal{C}^i$.

- *Verify*$(pk_{u,\mathcal{C}^i}, S, \tau, pk_{TTP}, \mathcal{C}^i) \rightarrow$(**true—false**) — is used to verify the validity of a pseudonym p. It is designed for checking that the pseudo-random pseudonym S and the pseudonym certificate τ were created by a valid e-token dispenser u to sign a freshly generated public key $pk_{u,\mathcal{C}^i}$ for a service context $\mathcal{C}^i$.
- *Identify*$(pk_{TTP}, S, \tau, \tau', pk_{u,\mathcal{C}^i}, pk'_{u,\mathcal{C}^i}) \rightarrow pk_u$ — is used to identify a user u that has generated multiple pseudonyms p to a given service context domain $\mathcal{C}^i$. Given two records of self-certified Sybil-free pseudonyms (S, τ) and (S, τ'), created by a user u when signing two different public keys $pk_{u,\mathcal{C}^i}$ and $pk'_{u,\mathcal{C}^i}$, $pk_{u,\mathcal{C}^i} \neq pk'_{u,\mathcal{C}^i}$, for the same service context $\mathcal{C}^i$, the algorithm *Identify* computes the public key pk_u of the owner of the e-token dispenser u. Thus, if a user generates more than one pseudonym for a given service context, it is possible to compute the public key pk_u that was used when requesting its initial identifier u (i.e., the e-token dispenser) to the TTP.
- *Revoke* $(sk_{TTP}, pk_{TTP}, r_u) \rightarrow pk'_{TTP}$ — is used by the TTP to revoke the initial identifier u. It takes as input the TTP's public and private key pair (pk_{TTP}, sk_{TTP}) and the revocation information r_u that is related to a particular user (see the *Obtain* algorithm). The *Revoke* algorithm outputs an updated issuer public key pk'_{TTP}. The dispenser u is revoked and can no longer be used to create signatures that verify this updated issuing key.

The algorithms *IKg* and *UKg*, *Obtain*, *Verify* and *Identify* are executed by the users $u \in \mathcal{U}$. The TTP, which issues initial identifiers u for the users, executes the algorithms *IKg* and *UKg*, *Issue*, *Verify*, *Identify*, and *Revoke*. In particular, *Verify* can be executed by any participant, including services $s \in \mathcal{S}$ that do not even need to possess a initial identifier u, or other third-party services that just monitor a service context to detect the presence of Sybil identifiers.

Further details regarding the algorithms used in the self-certified Sybil-free framework are found in Refs. 27), 30).

4. Identity Management Scheme

The identity management scheme, which we propose in this paper, is the point where the Internet of Service scenario, the trust model, and the self-certified Sybil-free identifiers come together (see also Fig. 1). There are four main steps in the proposed system: the bootstrapping, which is the initial step for any participant in the proposed system; the setting up of service contexts, which is usually performed by the service providers; the creation of pseudonyms for different service contexts; and the following use of such pseudonyms.

Bootstrapping: At first, we assume that each user and service provider who wants to participate in the system owns a unique, initial identifier, which is obtained at the bootstrapping phase from a party that is trusted by all involved parties (i.e., users, service directory provider, and service providers). For the service providers we assume that each provider is represented by the identifier obtained in this bootstrapping phase, i.e., each service provider is represented by a single identifier across all service contexts. It is also possible to create a pseudonym for each service provider per service context in the same way as for the users.

Setting up service contexts: In principle, any party can set up service contexts. In an Internet of Services scenario, it can be carry out by the party that publishes the directories with the different services or by a set of service providers that offer services with a similar nature. Setting up a service context requires a *unique identification tag* for each context. Such tags can be created from different sources, but for usability reasons they should at least provide a meaningful name for the service context, like MP3-downloads, online books, etc. A user-friendly

```
<ctx>
  <name_ctx>       Context Name       <\name_ctx>
  <valid_fr> 2009-08-01 08:00 GMT <\valid_fr>
  <valid_to> 2011-07-31 18:00 GMT <\valid_to>
  <region>            Europe             <\region>
  <rand_non> 4C656F6E6172646F414D <\rand_non>
  <init_pbK>        Public Key        <\init_pbK>
<\ctx>
```

Fig. 2 Example of a service context information $\mathcal{C}^i$.

option is to use an XML tag with context information, such as the name of the service context, region where the services are available, and validity time. The tag is then hashed into a unique value and used as input to the creation of the self-certified Sybil-free pseudonyms. **Figure 2** presents a context information $\mathcal{C}^i$ with 6 fields: the application name, starting time, expiration time, location, a nonce[⋆1], and the public key associated with the directory service or service provider or the pseudonym of the user that eventually set this service context.

Creating user pseudonyms: The pseudonyms of the users are bound to the service contexts and are created by the users themselves. User u_i issues a pseudonym $p_{u_i}^{\mathcal{C}^j}$ that is valid in the service context $\mathcal{C}^j$ using as input her identifier originally issued by a trusted third party, a freshly generated public-private key pair, and the unique information tag associated with the service context. The pseudonym is a tuple: newly generated public key, a serial number, and a certificate that proves the correctness of the operation (for details see Refs. 27), 30)).

Using the pseudonyms: Whenever a user u_i wants to interact with a service provider in a given service context $\mathcal{C}^j$, to evaluate the trustworthiness of a service provider in the context $\mathcal{C}^j$, or to provide recommendations in the context $\mathcal{C}^j$, she uses the pseudonym $p_{u_i}^{\mathcal{C}_j}$, which was created for this context. Thus, the real identity of the customers is not revealed to the service providers nor to other users. However, this still enables a service provider to recognize whether he has already interacted with a customer in the service context $\mathcal{C}^j$, and it enables other users to learn who is a trustworthy recommender in the context $\mathcal{C}^j$, as the customer has only a single pseudonym per context.

Expiration of pseudonyms & service contexts: A service context is valid until the validity time of the service context expires – if specified in its *unique identification tag*. When a service context expires, all pseudonyms bound to this service context become invalid. Users can also delete pseudonyms that are associate to them, but they are not able to create a new pseudonym for a service context that they were already part of – in such a case, users would need to restore the pseudonym that they had created for this service context before.

Furthermore, this pseudonym is also used for the exchange of recommendations about the behavior of the service providers between the users in the context $\mathcal{C}^j$. The differentiation between the trust relationships with regard to the different service contexts is as important for the application scenario since a user u_1 may trust u_2 in service context $\mathcal{C}^1$, but not in service context $\mathcal{C}^2$. In our approach, both users u_1 and u_2 are identified through their (unlinkable) pseudonyms $p_{u_1}^{\mathcal{C}^1}$, $p_{u_1}^{\mathcal{C}^2}$, $p_{u_2}^{\mathcal{C}^1}$, and $p_{u_2}^{\mathcal{C}^2}$ in the service contexts $\mathcal{C}^1$ and $\mathcal{C}^2$. Thus, the users can establish trust between each other and learn who (in the sense of the owner of which pseudonym) provides accurate recommendations. Furthermore, it's also possible to sign recommendations using the private key obtained during the creation of a pseudonym.

5. Security Evaluation

We assume that an attacker tries to manipulate the trust value of a service provider or to attack the users' privacy, i.e., the attacker aims to establish relationships between pseudonyms from different pseudonym sets associated with different service contexts. Hereby, we concentrate on the attacks which have a relation to the identity management scheme described in Section 4.

The attacker model allows attackers to participate in the system and to provide both misleading or correct recommendations to other users. The attacker can eavesdrop all communications between the service context and the pseudonym. We assume that the attacker can (try to) build relationships between pseudonyms only from the pseudonyms themselves, but not from the other sources of

[⋆1] A random nonce increases the entropy of the service context information and is used to prevent name collisions.

identification, such as the network layer information, i.e., IP addresses. Thus, we assume that an anonymous communication mechanism is used to link between users and services.

5.1 Attacks on Trust Systems

When the trust value of an entity is evaluated, the main factors that are considered are direct evidence and recommendations. This leads to two basic types of attacks. On the one hand, an entity can attack the model in the role of an interactor, e.g., it starts to build trust in order to exploit it later. This type of attack should be dealt with by the trust model itself, e.g., by considering the age of the evidence 24),34). As this attack has no relation to the identification scheme, we do not further evaluate it.

On the other hand, an attacker can try to influence a trust value in the role of a recommender, i.e., by providing misleading recommendations, either false praise or false accusation – again this type of attack should be dealt with in the trust model, e.g., by considering the trustworthiness of the recommenders 37). However, both kinds of attacks are susceptible to *whitewashing* 18), i.e., the attacker repeatedly joins the community as new entities in order to get rid of a bad history. Furthermore, an attacker can combine a Sybil attack 17) with the provision of misleading recommendations, in order to increase his overall impact on the trust value of a certain service provider. Here, an attacker would create an arbitrary high number of seemingly independent entities (i.e., different identifiers), which collusively provide misleading ratings for this service provider.

The proposed identity management scheme prevents both types of attacks. At first, whitewashing for service providers is not possible as they have only one identifier. This attack would also be prevented if service providers would be allowed to act pseudonymously per service context using the same type of pseudonym as the users. Whitewashing for recommenders is also not possible because a user is only allowed to have one pseudonym per context. If a user creates a second pseudonym, this can be detected given the underlying cryptographic construction, which allows the detection of multiple pseudonyms generated from a same user u_i to a given service context $\mathcal{C}^j$ by a pairwise comparison of the known pseudonyms 27),30). This, in turn, also means that Sybil attacks can be detected.

5.2 Pseudonym Unlinkability

The system architecture has strong unlinkability properties since the cryptographic properties of the k-times anonymous authentication ensure the algorithmic unlinkability of two pseudonyms $p_{u_1}^{\mathcal{C}^1}$, $p_{u_1}^{\mathcal{C}^2}$ generated for $\mathcal{C}^1$ and $\mathcal{C}^2$.

As f_s is a pseudo-random function, and all proof protocols are zero-knowledge, it is computationally infeasible to link the resulting e-token to a user, a dispenser u, or any other e-tokens corresponding to u. If a user shows two e-tokens in the same context domain to authenticate two pseudonyms p and p', then both e-tokens *must* use the same serial number.

The issuer, or any other participating device, can easily detect the violation and compute pk_u from the two double-show tags[⋆1]:

$$E = pk_u \cdot f_s(1\|\mathcal{C}^1)^{h(p)} \text{ and } E' = pk_u \cdot f_s(1\|\mathcal{C}^1)^{h(p')} \tag{1}$$

Thus, from Eq. (1), we have:

$$f_s(1\|\mathcal{C}^1) = \left(\frac{E}{E'}\right)^{(h(p)-h(p'))^{-1}} \quad \text{and} \tag{2}$$

$$pk_u = \frac{E}{f_s(1\|\mathcal{C}^1)^{h(p)}} = \frac{E'}{f_s(1\|\mathcal{C}^1)^{h(p')}} \tag{3}$$

However, the attacker may still be able to make an educated guess on whether two arbitrary pseudonym certificates from different identity domains are related or not, since information that may identify a device can be acquired from different sources in the TCP/IP stack, such as the network or application layers (thus, the initial assumption regarding the anonymous communication mechanism). In a real world scenario, additional information sources, like the geographical location of the user, could help the attacker to make such a guess.

6. Dealing with Sybil Attacks Efficiently

As described in Section 2.2 and Section 5.1, an adversary could try to increase her influence on the trust value of a certain service provider, by creating a seemingly high number of entities that provide misleading recommendations.

[⋆1] For a more detailed security analysis see Ref. 11).

Table 3 Notation: Efficient handling of Sybil attacks.

j	Rank of a recommender.
A	User who evaluates the trustworthiness of a service provider
C	Candidate for service provision under evaluation.
R_i	Recommender i.
(r_i, s_i)	Tuple of positive (r_i) and negative (s_i) units of evidence provided by recommender R_i.
t_i^A	The trustworthiness of recommender R_i in the context of providing recommendations (from A's point of view).
(r_{agg}, s_{agg})	Tuple of aggregated positive (r_{agg}) and aggregated negative (s_{agg}) units of evidence.
N_R	The maximum number of evidence units a recommender is expected to provide.
t_s	Threshold for Sybil attacks.
$imp(j)$	Aggregated impact of all recommenders with a rank equal or greater than j.
c	Constant defined as $c = (1 - t_s)N_R$.
f	Constant defining a fraction of c.
t_{imp}	Threshold for negligible impact ($t_{imp} = c/f$).
j_{neg}	Recommenders with a rank higher than j_{neg} will be neglected.

The proposed identity scheme allows for the detection of this type of attack by pairwise comparison of all identifiers which provide recommendations. This can potentially become a performance bottleneck of the aforementioned identity management scheme since it requires $\binom{n-1}{2}$ pairwise comparisons of identifiers to detect all Sybil identities. In this section we propose a solution to reduce the number of pairwise verifications by limiting the verification only to those identifiers that have a non-negligible influence on the trust value. An overview of the notation is provided in **Table 3** at the end of this section.

6.1 Basic Idea

Although decentralized trust systems, e.g., Refs. 10), 39), provide mechanisms to reduce the influence of a recommender based on his trustworthiness in the context of providing accurate recommendations, those models are usually susceptible to attacks where a single attacker creates a high number of seemingly independent recommenders, which she controls (Sybil attack), that then provide colluding misleading recommendations.

The approach we follow (based on Ref. 36)) reduces the impact of such an attack on the trust value by reducing the impact of each recommender not only based on the trustworthiness of this recommender in the context of providing accurate recommendations, but also on his rank j (1st most trusted recommender, 2nd most trusted recommender,…, jth most trusted recommender,…). This leads to a situation where the impact of lowly trusted recommenders which have a high rank (i.e., high value of j) can be neglected.

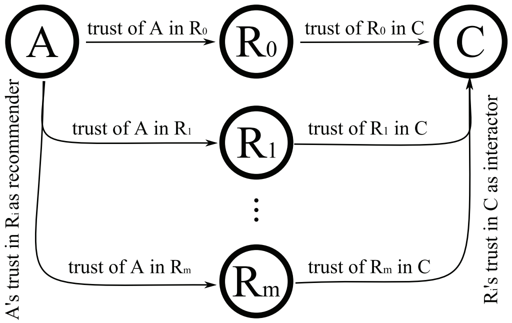

Fig. 3 Trust network.

In the following we show how this approach can be used to reduce the necessity to check the Sybil-freeness of all entities that provided recommendations, which is bounded by the cardinality of the user set.

6.2 Detailed Approach

In the following, we show how the number of entities that have a non-negligible influence on the trust evaluation depends on the trustworthiness of those entities and on their rank.

We assume there is a user u_1 that is going to evaluate the trustworthiness of a service provider, e.g., s_1, and furthermore user u_1 receives recommendations from multiple users $u_2, \ldots, u_m$. For simplifying the notation, we re-write this setting as follows: A user A that is going to evaluate the trustworthiness of a service provider C. Furthermore, user A receives recommendations from multiple recommenders $R_0, \ldots, R_m$ (see **Fig. 3**).

Each recommender R_i provides recommendations that describe C's past behavior from R_i's subjective point of view as a tuple of positive and negative evidence units (r_i, s_i), where r_i relates to the number of positive evidence units and s_i to the negative ones. Furthermore, A has information about the trustworthiness of each recommender R_i in the context of providing recommendations, which is denoted as $t_i^A \in [0, 1]$. The assessment of the trustworthiness can be based on a subjective (initial) belief about the trustworthiness of an entity and on the accuracy of past recommendations (see Ref. 37)). As a simplification, assume that the trustworthiness is used directly to reduce the impact of R_i's recommendations[⋆1]. Based on the trustworthiness t_i^A of the recommender R_i in the context of providing recommendations, it is possible to sort the recommenders R_i such that $t_i^A \geq t_{i+1}^A$. After this re-sorting the recommendations can be aggregated based on the following equation:

$$(r_{agg}, s_{agg}) = \Bigg(\sum_{i=0}^{m} \min\Big\{ t_i^A \cdot r_i, (1 - t_s) \cdot (t_i^A)^i \cdot \frac{N_R}{r_i + s_i} \cdot r_i \Big\}, \\ \sum_{i=0}^{m} \min\Big\{ t_i^A \cdot s_i, (1 - t_s) \cdot (t_i^A)^i \cdot \frac{N_R}{r_i + s_i} \cdot s_i \Big\} \Bigg) \tag{4}$$

Here, $t_s \in [0, 1]$ defines a threshold for Sybil attacks and N_R defines the maximum number of evidence units that entity A would ask for in order to believe that the collected information is representative for future behavior. Thus, we expect that for all i it also holds $r_i + s_i \leq N_R$ (otherwise r_i and s_i would be normalized to make the statement come true). The final trust value of the service provider C can be derived from the aggregated evidence as $(r_{agg} + 1)/(r_{agg} + s_{agg} + 2)$.

Finally, it can be shown that the impact on the aggregated evidence of the recommenders with rank greater than j, i.e., the recommenders R_j to R_m, is limited depending on j and t_j^A. We denote this impact as $imp(j)$; based on Eq. (4), it can be calculated as:

$$imp(j) = \sum_{i=j}^{m} \min\Big\{ t_i^A \cdot r_i, (1 - t_s) \cdot (t_i^A)^i \cdot \frac{N_R}{r_i + s_i} \cdot r_i \Big\} \\ + \sum_{i=j}^{m} \min\Big\{ t_i^A \cdot s_i, (1 - t_s) \cdot (t_i^A)^i \cdot \frac{N_R}{r_i + s_i} \cdot s_i \Big\} \tag{5}$$

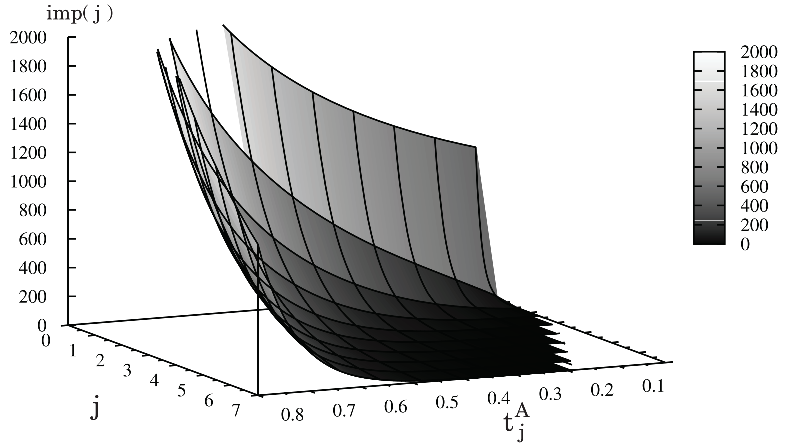

Fig. 4 Maximum impact of an infinite number of recommenders with a rank greater than j depending on j and t_j^A.

Based on this definition it can be shown that $imp(j)$ is bound by $(1 - t_s)N_R \sum_{i=j}^{m} (t_i^A)^i$ (see Ref. 35)). As $(1 - t_s)N_R$ are known before the evaluation they can be treated as constants and we define $c = (1 - t_s)N_R$. Using the properties of a geometric series it can be shown that this holds:

$$c \sum_{i=j}^{m} (t_i^A)^i \leq c \frac{(t_j^A)^j}{1 - t_j^A} \tag{6}$$

Here, $c(t_j^A)^j/(1 - t_j^A)$ is the maximum impact that could be achieved by an infinite number of entities (i.e., $m \to \infty$) with a rank greater than j and a trustworthiness not greater than t_j^A. **Figure 4** shows how $c \sum_{i=j}^{m} (t_i^A)^i$ for $m \to \infty$ evaluates depending on j and t_j^A for the constant $c = 1000$.

In Fig. 4, we can see that with a decrease of the trustworthiness t_j^A as well as with an increase of the rank j the impact $imp(j)$ of the recommenders with a rank greater than j falls rather quickly.

Next, we define the threshold $t_{imp} = c/f = (1 - t_s)N_R/f$ for negligible impact based on a fraction of the constant $c = (1 - t_s)N_R$, e.g., for $f = 1000$, $t_{imp} = c/1000$ holds. Given this threshold, we define the *set of recommenders with negligible impact* as the set of all recommenders with a rank greater than j_{neg}, where j_{neg} is defined by:

[⋆1] In the original proposal (see Ref. 36)) there is an additional function for deriving this weight (also called discounting factor) from the trustworthiness and additional parameters.

$$j_{neg} = \min\Big\{ j \in \mathbb{N} \mid t_{imp} \geq c\frac{(t_j^A)^j}{1 - t_j^A} \Big\} \tag{7}$$

which can be simplified to

$$j_{neg} = \min\Big\{ j \in \mathbb{N} \mid \frac{1}{f} \geq \frac{(t_j^A)^j}{1 - t_j^A} \Big\} \tag{8}$$

Based on this approach, we can reduce the number of entities that we consider when computing the trustworthiness of a service provider C to the number of recommenders that have a non-negligible impact, which in turn is well-suited to reduce the number of identifiers that have to be verified for being Sybil-free.

Example 1: Assuming that $f = 1000$ and for all $i \geq 11$ $t_i^A \leq 0.5$ holds, then $j_{neg} = 11$ and it is sufficient to consider the best 11 recommenders. This means that one has only to verify the Sybil-freeness of the identifiers of 11 recommenders, i.e., R_0,..., R_{10} (,and neglect the recommendations of all other recommenders).

Example 2: Assuming that $f = 1000$ and for all $i \geq 23$ $t_i^A \leq 0.7$ holds, then $j_{neg} = 23$ and it is sufficient to consider that best 23 recommenders. This means that one has only to verify the Sybil-freeness of the identifiers of 23 recommenders, i.e., $R_0, \ldots, R_{22}$ (and neglect the recommendations of all other recommenders).

7. Privacy for Services and Pseudonym Constructions

In this section we discuss the use of the aforementioned mechanisms for the provisioning of privacy for service providers and also summarize and evaluate other pseudonym constructions taking into account the requirements presented in Section 2.2.

7.1 Privacy for Services

The privacy of service providers in $\mathcal{S}$ can be protected using pseudonyms. The protection of the privacy of service providers is especially important when considering user-generated input, i.e., the role of individuals changes from service consumer to service provider. Unlinkability between service providers and service consumers can be obtained using anonymous communication mechanisms (see Ref. 28)) to disassociate senders to receivers. Application scenarios range from electronic forums to whistle-blowing sites and reporters of human rights abuses [29], where service is basically information input. Service contexts $\mathcal{C}$ are still used to group services of a similar nature. Hence, users providing content for different topics can belong to an arbitrary number of sets $\mathcal{S}^i$, where $1 < i < z$ and $z = |S|$ and, thus, to an equal number of service contexts $\mathcal{C}^i$.

The functionality of the proposed identity management scheme presented in Section 4 remains basically the same, with the exception that not only users, but also service providers can create pseudonyms. Services also benefit from the privacy-enhancing properties offered by the identity management scheme. Security and privacy properties of the system are not affected by such modifications.

7.2 Other Pseudonym Constructions

There are other pseudonym constructions that provide privacy-friendly identifiers. In this section, we summarize the most relevant pseudonym constructions that are related to our proposal. Such proposals are based on different signatures schemes, identity-based encryption, pairing, and semantically secure encryption.

Anonymous authentication providing unlinkability between multiple appearances of the same identifier can be implemented using group signatures [6],[15]. However, group signature schemes alone do not provide any protection against a signer generating any two group signatures, i.e., the deployment of Sybil identifiers [16],[41].

Identity-based encryption schemes can be used to construct pseudonyms. The pseudonym-based encryption scheme proposed in Ref. 22) is based on pairings and constructed on top of an identity-based encryption scheme [7] and short signatures from the Weil pairing [8]. The main disadvantage of the pseudonym-based encryption scheme is that it is vulnerable to a Sybil attack. Any device with an initial identifier, i.e., a public key is used as the initial identifier in this scheme, can generate an arbitrary number of pseudonyms.

Anonymous X.509 attribute certificates can be constructed using different signature schemes, such as fair blind signatures, traceable signatures, and ring signatures [3]. Attribute certificates created with fair blind signatures [38] were presented in Ref. 4). However, such schemes do not provide unlinkability between multiple appearances of the same attribute certificate. A traceable signature scheme [26], which is a group signature scheme with additional tracing capabilities [3], can be used as a cryptographic primitive to set up privacy-friendly X.509 attribute certificates that can provide unlinkability between different appearances of the same attribute certificate[5].

Semantically secure encryption can be used to generate changeable pseudonyms[43]. Changeable transaction pseudonyms, which are represented by ciphertexts, were presented in Ref. 42). It is possible to associate changeable transaction pseudonyms with a history of events and use them with a reputation mechanism based on events[42]. However, the use of semantically secure encryption schemes alone cannot offer protection against the deployment of Sybil identifiers.

8. Conclusions

In this paper, we presented an identity management scheme for a service environment enabling users to establish trust, yet, preserving a user's privacy. The trade-off between privacy (asking for anonymity at best) and trust (requiring for long-term identifiers) is set by defining how each user is able to have only one pseudonym per context. However, those identifiers cannot be linked across service contexts.

The services context are linked to services that are similar in nature; they can be directly derived from a service directory. The proposed identity management scheme supports the establishment of trust within each of those service contexts. Furthermore, when the trust model takes recommendations from third parties into account (as in e.g., Refs. 36), 39)), then a user can also learn whether the recommendations from a certain recommender tend to be correct or misleading.

In general, trust models benefit from the proposed identifier scheme as the construction of the identifiers aids in the detection of Sybil attacks, which are considered to be a major threat to many trust models in distributed systems. As the number of verifications required for detecting a Sybil attack can become a bottleneck, we introduced a new approach for reducing the number of entities that are considered in the trust evaluation.

Furthermore, we have presented how services can also benefit from the proposed identity management scheme and have also listed some pseudonym constructions, and their disadvantages in relation to our scheme.

Finally, the users' privacy is preserved so that a users' behavior cannot be tracked across the boundaries of contexts, but it can be tracked within contexts. This allows service providers to create a history of its customers in a certain context, e.g., music, which is clearly preferable for the service providers, as they can tailor their advertising to the profile of the customer. However, a service provider has no means to recognize a customer in another context. Especially, we like to emphasis, that due to the construction of the identifiers, the Sybil-freeness of any set of identifiers (within a given context) can be verified without the need for an online certification authority.

References

1) Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data, Official Journal L No.281 (1995). See http://www.cdt.org/privacy/eudirective/EU_Directive_.html

2) Andersson, C., Kohlweiss, M., Martucci, L.A. and Panchenko, A.: A Self-Certified and Sybil-Free Framework for Secure Digital Identity Domain Buildup, *Information Security Theory and Practices: Smart Devices, Convergence and Next Generation Networks, Proc. 2nd IFIP WG 11.2 International Workshop* (*WISTP 2008*), Onieva, J.A., Sauveron, D., Chaumette, S., Gollmann, D. and Markantonakis, K. (Eds.), Lecture Notes in Computer Science, LNCS 5019, pp.64–77, Springer (2008).

3) Benjumea, V., Choi, S.G., Lopez, J. and Yung, M.: Anonymity 2.0: X.509 Extensions Supporting Privacy-Friendly Authentication, *CANS 2007, 6th International Conference on Cryptography and Network Security*, Bao, F. and Okamoto, T. (Eds.), Lecture Notes in Computer Science, Vol.4856, pp.265–281, Springer-Verlag (2007).

4) Benjumea, V., Lopez, J., Montenegro, J.A. and Troya, J.M.: A First Approach to Provide Anonymity in Attribute Certificates, *Proc. 7th International Workshop on Practice and Theory in Public Key Cryptography* (*PKC 2004*), Bao, F., Deng, R.H. and Zhou, J. (Eds.), Lecture Notes in Computer Science, Vol.2947, LNCS 2974, pp.402–415, Springer-Verlag (2004).

5) Benjumea, V., Lopez, J. and Troya, J.M.: Anonymous Attribute Certificates based on Traceable Signatures, *Internet Research: Electronic Networking Applications and Policy. Special Issue on Privacy and Anonymity in the Digital Era: Theory, Technologies and Practice*, Vol.16, No.2, pp.120–139 (2006).

6) Boneh, D., Boyen, X. and Shacham, H.: Short Group Signatures, *Proc. 24th Annual International Cryptology Conference on Advances in Cryptology* (*CRYPTO 2004*), Franklin, M.K. (Ed.), Lecture Notes in Computer Science, Vol.3152, pp.41–55, Springer (2004).

7) Boneh, D. and Franklin, M.K.: Identity-Based Encryption from the Weil Pairing, *SIAM Journal of Computing*, Vol.32, No.3, pp.586–615 (2003).

8) Boneh, D., Lynn, B. and Shacham, H.: Short Signatures from the Weil Pairing,

Proc. 7th International Conference on the Theory and Application of Cryptology and Information Security, Advances in Cryptology (*ASIACRYPT 2001*), Boyd, C., (Ed.), Lecture Notes in Computer Science, Vol.2248, pp.514–532, Springer (2001).

9) Brickell, E.F., Camenisch, J. and Chen, L.: Direct anonymous attestation, *Proc. 11th ACM Conference on Computer and Communications Security* (*CCS 2004*), Atluri, V., Pfitzmann, B. and McDaniel, P.D. (Eds.), pp.132–145, ACM (2004).

10) Buchegger, S. and Boudec, J.-Y.L.: A Robust Reputation System for Peer-to-Peer and Mobile Ad-hoc Networks, *P2PEcon 2004* (2004).

11) Camenisch, J., Hohenberger, S., Kohlweiss, M., Lysyanskaya, A. and Meyerovich, M.: How to Win the Clone Wars: Efficient Periodic n-Times Anonymous Authentication, *Proc. 13th ACM Conference on Computer and Communications Security* (*CCS 2006*) (2006).

12) Camenisch, J. and Lysyanskaya, A.: A Signature Scheme with Efficient Protocols, *Security in Communication Networks: 3rd International Conference* (*SCN 2002*), Lecture Notes in Computer Science, Vol.2576/2003, pp.268–289, Springer, Amalfi, Italy (2002).

13) Cameron, K.: The Laws of Identity, www.identityblog.com (2005).

14) Cavoukian, A.: 7 Laws of Identity: The Case for Privacy-embedded laws of identity in the digital age, Whitepaper (2006).

15) Chaum, D. and van Heyst, E.: Group Signatures, *Proc. Workshop on the Theory and Application of of Cryptographic Techniques, Advances in Cryptology* (*EUROCRYPT 1991*), Davies, D.W. (Ed.), Lecture Notes in Computer Science, Vol.547, pp.257–265, Springer (1991).

16) Defrawy, K.E. and Tsudik, G.: ALARM: Anonymous Location-Aided Routing in Suspicious MANETs, *Proc. 15th IEEE International Conference on Network Protocols* (*ICNP 2007*), pp.304–313, IEEE (2007).

17) Douceur, J.R.: The Sybil Attack, *Peer-to-Peer Systems: Proc. 1st International Peer-to-Peer Systems Workshop* (*IPTPS*), Druschel, P., Kaashoek, F. and Rowstron, A. (Eds.), Vol.2429, pp.251–260, Springer-Verlag (2002).

18) Feldman, M. and Chuang, J.: Overcoming free-riding behavior in peer-to-peer systems, *SIGecom Exch.*, Vol.5, No.4, pp.41–50 (2005).

19) Fiat, A. and Shamir, A.: How to Prove Yourself: Practical Solutions to Identification and Signature Problems, *Proc. Advances in Cryptology* (*CRYPTO 1986*), Odlyzko, A.M. (Ed.), Lecture Notes in Computer Science, Vol.263, pp.186–194, Springer (1987).

20) Gal-Oz, N., Gilboa, N. and Gudes, E.: Schemes for Privately Computing Trust and Reputation, *Proc. 4th IFIP International Conference on Trust Management* (*IFIPTM 2010*), pp.1–16, Springer Boston (2010).

21) Goldwasser, S., Micali, S. and Rivest, R.L.: A digital signature scheme secure against adaptive chosen-message attacks, *SIAM J. Comput.*, Vol.17, No.2, pp.281–308 (1988).

22) Huang, D.: Pseudonym-based cryptography for anonymous communications in mobile ad hoc networks, *International Journal of Security and Networks* (*IJSN*), Vol.2, No.3/4, pp.272–283 (2007).

23) Jøsang, A.: Trust-based Decision Making for Electronic Transactions, *Proc. 4th Nordic Workshop on Secure IT Systems* (*NORDSEC'99*), Yngström, L. and Svensson, T. (Eds.) (1999).

24) Jøsang, A. and Ismail, R.: The Beta Reputation System, *Proc. 15th Bled Conference on Electronic Commerce* (2002).

25) Jøsang, A., Ismail, R. and Boyd, C.: A Survey of Trust and Reputation Systems for Online Service Provision, *Decision Support Systems*, Vol.43, No.2, pp.618–644 (2007).

26) Kiayias, A., Tsiounis, Y. and Yung, M.: Traceable Signatures, *Proc. 23th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Advances in Cryptology* (*EUROCRYPT 2004*), Cachin, C. and Camenisch, J. (Eds.), Lecture Notes in Computer Science, Vol.3027, pp.571–589, Springer (2004).

27) Martucci, L.A.: Identity and Anonymity in Ad Hoc Networks, PhD Thesis, Karlstad University (2009).

28) Martucci, L.A., Andersson, C. and Fischer-Hübner, S.: Chameleon and the Identity-Anonymity Paradox: Anonymity in Mobile Ad Hoc Networks, *Proc. 1st International Workshop on Security* (*IWSEC 2006*), Information Processing Society of Japan (IPSJ), pp.123–134 (2006).

29) Martucci, L.A. and Fischer-Hübner, S.: Privacy for Reporters of Human Rights Abuses, *Mobile Technologies for Conflict Management, New Avenues for Online Dispute Resolution*, Poblet, M. (Ed.), Law, Governance and Technology Series, first edition, Springer (2011).

30) Martucci, L.A., Kohlweiss, M., Andersson, C. and Panchenko, A.: Self-Certified Sybil-Free Pseudonyms, *Proc. 1st ACM Conference on Wireless Network Security* (*WiSec'08*), pp.154–159, ACM Press (2008).

31) Merwe, J.V.D., Dawoud, D. and McDonald, S.: A survey on peer-to-peer key management for mobile ad hoc networks, *ACM Computing Surveys*, Vol.39, No.1, pp.1–45 (2007).

32) Pfitzmann, A. and Hansen, M.: A terminology for talking about privacy by data minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management v.0.34 (2010).

33) Quercia, D. and Hailes, S.: MATE: Mobility and Adaptation with Trust and Expected-utility, *International Journal of Internet Technology and Secured Transactions* (*IJITST*), Vol.1, pp.43–53 (2007).

34) Ries, S.: Extending Bayesian Trust Models Regarding Context-Dependence and User Friendly Representation, *Proc. 2009 ACM Symposium on Applied Computing*, ACM Press (2009).

35) Ries, S.: Trust in Ubiquitous Computing, PhD Thesis, Technische Universität Darmstadt (2009).
36) Ries, S. and Aitenbichler, E.: Limiting Sybil Attacks on Bayesian Trust Models in Open SOA Environments, *Proc. 1st International Symposium on Cyber-Physical Intelligence* (*CPI-09*) (2009).
37) Ries, S. and Heinemann, A.: Analyzing the Robustness of CertainTrust, *2nd Joint iTrust and PST Conference on Privacy, Trust Management and Security*, Karabulut, Y., Mitchell, J.C., Herrmann, P. and Jensen, C.D. (Eds.), pp.51–67, Springer (2008).
38) Stadler, M., Piveteau, J.-M. and Camenisch, J.: Fair Blind Signatures, *Proc. International Conference on the Theory and Application of Cryptographic Techniques, Advances in Cryptology* (*EUROCRYPT 1995*), Guillou, L.C. and Quisquater, J.-J. (Eds.), Lecture Notes in Computer Science, Vol.921, pp.209–219, Springer (1995).
39) Teacy, W.T.L., Patel, J., Jennings, N.R. and Luck, M.: TRAVOS: Trust and Reputation in the Context of Inaccurate Information Sources, *Autonomous Agents and Multi-Agent Systems*, Vol.12, No.2, pp.183–198 (2006).
40) Teranishi, I., Furukawa, J. and Sako, K.: k-Times Anonymous Authentication (Extended Abstract), *Proc. 10th International Conference on the Theory and Application of Cryptology and Information Security, Advances in Cryptology* (*ASIACRYPT 2004*), Lecture Notes in Computer Science, Vol.3329, pp.308–322, Springer (2004).
41) Tsudik, G. and Xu, S.: A Flexible Framework for Secret Handshakes, *Revised Selected Papers of the 6th International Workshop Privacy Enhancing Technologies* (*PET 2006*), Danezis, G. and Golle, P. (Eds.), Lecture Notes in Computer Science, Vol.4258, pp.295–315, Springer (2006).
42) Weber, S.G., Martucci, L.A., Ries, S. and Mühlhäuser, M.: Towards Trustworthy Identity and Access Management for the Future Internet, *Proc. 4th International Workshop on Trustworthy Internet of People, Things & Services* (*IoPTS*) (2010).
43) Weber, S.G. and Mühlhäuser, M.: Multilaterally Secure Ubiquitous Auditing, *Intelligent Networking and Collaborative Systems and Applications, Studies in Computational Intelligence*, Vol.329, Springer (2010).
44) Whitby, A., Jøsang, A. and Indulska, J.: Filtering Out Unfair Ratings in Bayesian Reputation Systems, *The ICFAIN Journal of Management Research*, Vol.4, No.2, pp.48–64 (2005).
45) Zimmermann, P.R.: *The Official PGP User's Guide*, MIT Press, Cambridge, MA, USA (1995). Out of Print.

(Received November 1, 2010)
(Accepted April 8, 2011)
(Released July 6, 2011)

Leonardo A. Martucci was born in 1977. He is a principal investigator at the Center for Advanced Security Research Darmstadt (CASED), where he leads the Informational Privacy Research Group since 2010. He is also a post-doctoral researcher at the Telecooperation Lab (TK), Technische Universität Darmstadt, Germany, since 2009. He received his Doctor and Licentiate Engineering degrees in Computer Science from Karlstad University, Sweden, in 2009 and 2006, respectively, and his Masters and Diploma in Electrical Engineering degrees from the University of São Paulo, Brazil, in 2002 and 2000 respectively. His research interests include network security and privacy enhancing technologies in general, with a focus on wireless and decentralized systems. His current research topics are privacy, smart grids, and digital identities.

Sebastian Ries was born in 1979. He is the coordinator of the research area Secure Services at the Center for Advanced Security Research Darmstadt (CASED) since 2009. He is the head of the research group Smart Security & Trust at the Telecooperation Lab (TK), Technische Universität Darmstadt, Germany, since 2008, and a principle investigator at CASED since 2010. He obtained his Doctor and Diploma degrees in Computer Science from Technische Universität Darmstadt obtained in 2009 and 2005, respectively. He was awarded with research scholarships by the German National Science Foundation (DFG) and CASED, while preparing his dissertation. His research interests include trust and reputation models, trust establishment in complex systems, as well as challenges in the fields of privacy and usable security.

Max Mühlhäuser was born in 1957. He is a full professor at the Technische Universität Darmstadt, Germany, where he heads the Telecooperation Lab (TK) since 2000. He is also a directorate member of the Center for Advanced Security Research Darmstadt (CASED) where he is heading the Secure Services division. He received his Doctor and Diploma degrees from Universität Karlsruhe, Germany, in 1986 and 1981 respectively. He started and managed an industrial research center and held permanent or visiting professorships at the Universities of Kaiserslautern, Karlsruhe, Linz, Darmstadt, Montréal, Sophia Antipolis (Eurecom), and San Diego (UCSD). In 1993, he founded the TeCO institute in Karlsruhe, Germany, which became one of the pace-makers for ubiquitous computing research in Europe. His current research interests in the security area are pervasive and civil security, usability, privacy and trust.