University of Murcia

Faculty of Computer Science

Department of Information
and Communications Engineering

PhD Thesis

Enhancing User-Centric Identity Management Systems with Reputation Models in Distributed Environments

Mejora de Sistemas de Gestión de Identidades Centrados en el Usuario mediante Modelos de Reputación en Entornos Distribuidos

Author
Ginés Dólera Tormo

Thesis advisors
Dr. Félix Gómez Mármol
Prof. Dr. Gregorio Martínez Pérez

Murcia, October 2014

The following Thesis is a compilation of the next published articles, being the PhD student the main author in all of them:

1. Ginés Dólera Tormo, Félix Gómez Mármol, Gregorio Martínez Pérez, "**Identity Management in Cloud Systems**", Security, Privacy and Trust in Cloud Systems, Eds: S. Nepal, M. Pathan, Publisher: Springer, ISBN: 978-3-642-38585-8, pp 177-210, 2014
`http://dx.doi.org/10.1007/978-3-642-38586-5_6`

2. Ginés Dólera Tormo, Félix Gómez Mármol, Gregorio Martínez Pérez, "**On the Application of Trust and Reputation Management and User-centric Techniques for Identity Management Systems**", XII Spanish Meeting on Cryptology and Information Security (RECSI 2012), San Sebastián, Spain, 04-07 September 2012

3. Ginés Dólera Tormo, Félix Gómez Mármol, Gregorio Martínez Pérez, "**Towards the Integration of Reputation Management in OpenID**", Computer Standards & Interfaces, Special Issue on Secure Mobility in Future Communication Systems under Standardization, vol. 36, no. 3, pp. 438-453, March 2014
`http://dx.doi.org/10.1016/j.csi.2013.08.018`

4. Ginés Dólera Tormo, Félix Gómez Mármol, Gregorio Martínez Pérez, "**ROMEO: ReputatiOn Model Enhancing OpenID Simulator**", 19th European Symposium on Research in Computer Security (ESORICS), Security & Trust Management Workshop (STM 2014), Wroclaw, Poland, 7-11 September 2014. LNCS 8743, pp. 193-197
`http://dx.doi.org/10.1007/978-3-319-11851-2_15`

5. Ginés Dólera Tormo, Félix Gómez Mármol, Joao Girao, Gregorio Martínez Pérez, "**Dynamic and Flexible Selection of a Reputation Mechanism for Heterogeneous Environments**", Future Generation Computer Systems, Special Issue on Trustworthy Data Fusion and Mining in Internet of Things, 2014
`http://dx.doi.org/10.1016/j.future.2014.06.006`. (In press)

6. Ginés Dólera Tormo, Félix Gómez Mármol, Joao Girao, Gregorio Martínez Pérez, "**Identity Management: In privacy we trust. Bridging the trust gap in e-Health environments**", IEEE Security & Privacy, Special Issue on Health IT Security and Privacy, vol. 11, no. 6, pp. 34-41, 2013
`http://dx.doi.org/10.1109/MSP.2013.80`

7. Ginés Dólera Tormo, Félix Gómez Mármol, Gregorio Martínez Pérez, "**Towards privacy-preserving reputation management for hybrid broadcast broadband applications**", Computers & Security, 2014. (Accepted)

Table of Contents

Acknowledgements

Agradecimientos

En primer lugar, quisiera dar las gracias a mi familia. A mis padres, por su apoyo incondicional, por haber confiado siempre en mí y por no dejar que me rinda nunca. Porque no habría llegado tan lejos si no fuera por ellos. Por su paciencia, por los valores que me han enseñado, y por ser un constante ejemplo a seguir. Porque aunque ellos digan que están orgullosos de mí, más lo estoy yo de ellos. A mi hermano Juan, que nunca ha dudado de que llegaría donde me propusiera. Y sobre todo, agradecer a mi abuela Paqui, que me sigue ayudando aún hoy día.

Quisiera también agradecer a Clara Eugenia, por su alegría, su sonrisa y su cariño. Porque siempre consigue hacerme ver el lado positivo de todo, y por permanecer a mi lado a lo largo de toda la tesis. Porque me hace sentir un hombre afortunado.

No podría olvidarme de agradecer a mis amigos de toda la vida, por hacerme sentir como si el tiempo y la distancia no importaran, y a los de salsa, por su cariño y alegría. Son muchos para nombrarlos a todos, pero ellos saben quiénes son. A los de la facultad, en especial a Reyes, Jose, Manolo y Álex, por tantos buenos momentos, a los de Heidelberg, en espacial a Dennis, Wenting, Jens y Brigitta por hacer el día a día tan agradable. Por supuesto agradecer a Pedro, Fran y Andrés por alegrar tantos momentos de trabajo, y por todas las risas que hemos compartido y sin olvidar al gran Manuel Gil. También quería dar las gracias a Gabriel López, ya que aprendí mucho de él y con él en los duros comienzos. A Joao Girao, por darme la oportunidad de aprender a su lado y por su sentido del humor.

Finalmente, un agradecimiento especial merecen mis directores de tesis. A Gregorio, por darme mi primera oportunidad profesional, confiando en mí desde el principio, por su sencillez y cercanía, con la que nunca he tenido la sensación de relación jefe-empleado, sino de compañeros y amigos. Por siempre anteponer mis intereses a los suyos, por su paciencia y mano izquierda, y por siempre poner la voz de la experiencia ante cualquier duda. A Félix, excepcional profesional pero mejor persona, por haberme guiado en este duro proceso, incluyendo nuestras interminables discusiones, porque gracias a él he avanzado en mi carrera profesional a pasos agigantados, pero sobre todo, por la ilusión que pone en todo lo que hace, por su disponibilidad a echar una mano siempre que ha hecho falta y por esa humildad con la que me ha tratado desde que nos conocimos. Ambos me han enseñado profesionalmente, pero sobre todo agradezco lo que he aprendido de ellos a nivel personal. Sin duda, este trabajo no habría sido posible sin contar con su apoyo.

A todos ellos, muchas gracias.

Acknowledgements

Firstly, I would like to thank my family. My parents, for their unconditional support, for their permanent trust in me and not letting me to give up ever. Because I would not have come this far without them. For their patience, the values they have taught me, and for being a constant example to follow. Even though they say they are proud of me, I am more proud of them. To my brother Juan, who has never doubted that I would come where I wanted. And special thanks to my grandmother Paqui, who is helping me even nowadays.

I would also want to thank Clara Eugenia, for her joy, her smile and her affection. Because she always makes me see the positive side of everything, and for staying with me throughout the whole thesis. Because she makes me feel lucky.

I cannot forget to thank my lifelong friends, for making me feel like time and distant do not matter, and my friends of salsa, for their joy and affection. There are too many to name them all, but they know who they are. To my faculty friends, especially Reyes, Jose, Manolo and Álex, for so many good times, to my Heidelberg friends, especially Dennis, Wenting, Jens and Brigitta for making a pleasant daily life. Of course thanks to Pedro, Fran y Andrés to cheer many moments at work, and all the laughs we shared, and not forgetting the great Manuel Gil. I also want to thank Gabriel López, since I learnt a lot from him and with him in the tough beginnings. To Joao Girao, for giving me the opportunity to learn by his side and for his sense of humor.

Finally, special thanks to my thesis advisors. To Gregorio, for giving me my first professional opportunity, trusting me from the very first moment, for his naturalness and proximity, with which I never had the feeling of a boss-employee relationship, but rather colleagues and friends. For always putting my interests ahead of his own ones, for his patience and tact, and always providing the voice of experience on any doubts. To Félix, exceptional professional yet better person, for guiding me in this hard process, including our endless discussions, because thanks to him I have advanced in my career by leaps and bounds, but above all, for the illusion that he puts on everything he does, for his willingness to lend a hand whenever it has been needed, and for the humbleness with which he has treated me since we met. Both have taught me professionally, but I especially appreciate what I have learnt from them personally. Definitely, this work would not have been possible without their support.

To all of them, thank you very much.

Abstract

I Motivation and Goals

Due to the great success of communication systems in the Internet in the last years, its users are increasing the amount of exchanged information, among which sensitive and personal data are included. However, these users sometimes are not aware enough of how their personal data are being handled, or they do not know who can actually have access to such information and with which purposes.

It is fairly common nowadays to deal with sign up forms even for services which will be likely used only once, e.g., commenting an entry in a blog. Data asked for using these services is usually personal, like email address or birthdate; sometimes even more private data is requested, which might seem irrelevant for the provision of the service itself, such as phone number or hobbies.

Such unnecessary creation of user accounts (with its associated collection of sensitive data) not only results in having to remember many different usernames and passwords for each service, or expose ourselves to receive spam, but it also jeopardizes the privacy of the users. In many cases, when users provide their personal data, they do not really know how this data is going to be handled, who will this data be released to, or whether it could be used for marketing campaigns outside the service they are signing up, for instance.

Gathering knowledge about users is considered more and more appealing, becoming even a target for some organizations with commercial purposes. The collected information is mostly used within advertisement goals, or it is aimed to develop advanced attacks on specific targets extracted from such acquired data. Several services offered in the Internet are willing to collect information of the users, either explicitly, for instance using the aforementioned sign up forms, or by other means, such as trying to infer users' profiles through the interaction they perform over the offered services.

Both privacy protection and control over the information collected by other entities about oneself are characteristics more and more sought by the users of any communication system. Besides, these issues are considered a right of the users in certain geopolitical environments, such as the European Union. In this context, we can find users that do not want to relate their private life as they interact with different web services, or users that want to avoid the services to collect information about their preferences or to build usage profiles. For instance, journalists willing to report controversial events without being concerned about potential reprisals, militaries that cannot reveal their geographical location, or just as an additional security measure for any user of the communication systems in the Internet.

An elegant solution to these concerns has begun to spread recently through the usage of the identity management systems. Identity management systems establish trust relationships between different organizations, in such a way that the information of the users is handled by a trusted entity, usually known as identity provider (for example, their city council, university, Internet access provider, known email providers or any other trustworthy organization).

When users access a service in the Internet (e.g., an online shopping site, subscription to a news feed, commenting an entry in a blog, etc.), the authentication methods and users' data management is delegated to the identity provider. Thus, the identity provider prevents the users from signing up and sending their private information directly to the services they want to use. The service only gets the information sent by the identity provider, which may hide the real identity of the users, hence preserving their privacy. Additionally, identity management systems provide Single Sign-On, since they allow the users to make use of a unique account (the one they have in the identity provider) to access different services repeatedly. Moreover, identity management systems are adopting the user-centric paradigm, which embraces usability and a higher focus on users' needs as key drivers.

Diverse solutions and standards have been developed in a number of initiatives, in order to define the communication between identity providers and Internet services. As representative examples, we can consider SAML or OpenID. However, these systems present numerous shortcomings even nowadays, especially in the aspects related to the control that the users have on their own information. Current solutions barely inform the users on how their personal information is handled, and they rarely allow the users to decide which data may or may not be transferred to other entities.

Furthermore, in those cases where the users are informed about which services are requesting their personal data, they do not know how much they can trust those services. In other words, the users have no means to know how their information is going to be processed, or if the requested service will fulfill their expectations.

These difficulties are even further increased when identity management systems have to be deployed in distributed environments, such as P2P networks, where trust relationships cannot be established through static centralized servers. In such cases, establishing static contracts in order to set up the quality of service between a service provider and the clients, such as Service Level Agreements (SLA), or to determine how private information is handled is hardly applicable. Such limitation raises the necessity of deploying additional mechanisms to manage trust.

As an alternative to manage trust in distributed environments, reputation management systems have been successfully applied in recent years. In these systems, the trust that can be placed on a given entity (such as a network node, a service, or even a user), is not established by static agreements, but it is rather based on past experiences that others have been having with such an entity.

Reputation management systems attempt to predict the behavior of an entity from the behavior that such an entity had in the past. The reputation is usually computed from the feedback provided by other entities or users that already interacted with the given entity. In this way, when a user wants to interplay with an unknown entity, the user can be informed about the behavior in the past of this entity, deciding whether continuing or not with the interaction.

Even though both identity management and reputation management have been positively developed, merging both worlds is not straightforward in distributed environments, since it raises a number of challenges that need to be taken into consideration. For instance, some solutions have been proposed in order to endow the system defined by the OpenID standard with reputation management mechanisms. Nevertheless, those solutions are based on establishing trustworthy and static centralized services, actually not fitting in distributed environments which, indeed, OpenID is aimed to.

In distributed environments, it is not only that trustworthy centralized services are not applicable, but also that any user can participate using several identities and deploy its own services. If appropriate measures are not undertaken, collaborative attacks might be introduced, where a vast amount of nodes could be deployed with malicious purposes, such as unfairly increasing or decreasing the reputation of other entities.

Moreover, system conditions could be highly volatile in terms of amount of users, amount of deployed nodes, their participation, amount of malicious users, etc. Such scenario requires the reputation systems to be highly dynamic, in a way that they should be able to adapt to the changes of the environments where they are deployed.

Additionally, in order to make the reputation management systems work, the users have to supply recommendations about the services they have been using. In this way, those recommendations are aggregated by using different mechanisms, which could even provide customized reputation values taking into account the users' preferences.

However, that would imply the service in charge of aggregating those recommendations to know the recommendations supplied by each user, the service they have been accessing and even their preferences. That would be against what user-centric identity management systems want to preserve, that is, the privacy of the users.

Due to the challenges of enhancing user-centric identity management systems with reputation models in distributed environments, the goals pursued within this thesis are the following:

- Study the current state of the art of identity management systems and related standards, analyzing open issues, research lines and ongoing work that this kind of systems pose.
- Analyze the current state of the art regarding reputation management systems applicable to identity management systems scenarios.
- Identify the advantages and disadvantages as well as the difficulties and challenges that the integration of reputation management systems with user-centric identity management systems raises in distributed environments.
- Design and suggest solutions allowing the integration of reputation management systems with current user-centric identity management systems in distributed environments, in such a way that the users could assess whether a service provider will fulfill their expectations before interacting with it.
- Perform a deep analysis of the behavior of such solutions, making use of different mechanisms to aggregate recommendations, and considering malicious users and entities, too.
- Propose and analyze solutions aimed to enhance the adaptability of the current reputation management systems in dynamic environments.
- Explore and propose solutions to improve the privacy of current reputation management systems within the context of user-centric identity management systems.

II Methodology

This PhD dissertation sets its starting point at analyzing the state of the art on identity management systems (Chapter 1). As part of this analysis, we identified essential requirements these systems may fulfill after studying different relevant use cases and scenarios within the context of user-centric identity management solutions.

Our first contribution consists of describing representative identity management standards, together with related technologies and approaches, in order to highlight the benefits and drawbacks of them, according to the identified requirements, focusing on user-centric aspects. An interesting result showed that there is not an optimal solution able to fulfill all requirements, which emphasizes that there are still important open issues in this field.

One of the most interesting results remarks the lack of trust management that identity management systems present in distributed environments. The identity management systems we analyzed usually assume trust relationships between entities, in some cases enforced by static service level agreements. Nevertheless, this way of managing trust is hardly applicable in more dynamic environments.

To bridge this gap, we focused on an alternative that has been successfully deployed to manage trust in distributed environments, namely, reputation management systems. Thus, it seemed reasonable to think that such shortcomings could be addressed by reputation management systems.

Thereby, we made a deep analysis of the state of the art on reputation management systems too, within the context of user-centric identity management systems, and taking the intrinsic characteristics of distributed environments as a reference. The idea was to improve user-centric identity management systems in order to properly inform the users before enjoying the services they want to consume, including environments where trust cannot be handled with static agreements.

Nevertheless, we soon realized that the integration between both fields was not straightforward, and numerous considerations had to be taken into account for a proper execution (Chapter 2). Most of the existing works in that direction propose mechanisms to manage reputation by using centralized services, being hardly applicable in more dynamic or distributed contexts.

Thus, we decided to design a reputation management model (Chapter 3) applied to a user-centric identity management system, distributed and currently widely spread, which is the case of OpenID. To the best of our knowledge, this was one of the first solutions proposing a reputation management system applied to such a standard.

As part of the analysis of the aforementioned solution, we proposed a number of mechanisms to aggregate the collected users' recommendations, differing in its complexity, requirements to make them work, and capabilities to prevent ill-intentioned users, among other features. In this way, we did not only analyze whether the proposed solution is feasible, but also different ways of computing the reputation values, whose behavior depends on the system conditions (e.g. number of users, users' participation, percentage of malicious users, etc.) and the expected performance measurements (e.g., computational resources, network resources, accuracy in the reputation values, etc.).

For a proper study of the previous solution, we carried out the implementation of a simulator able to depict the behavior of the proposed reputation management model against different scenarios (Chapter 4). As part of this work, we described threats which may compromise reputation management models in that environment. From these threats, we identified the scenario elements to be modeled, including different kind of malicious users and entities. The simulator implements several mechanisms for aggregating recommendations, as well as different charts in order to visualize the simulation results from various perspectives. In this way, the behavior of each of the mechanisms of the reputation model previously described can be analyzed against diverse scenarios.

The results of such analysis were positive in order to demonstrate the feasibility of the model. However, the analysis of the distinct mechanisms used to aggregate users' recommendations made us realize the variability of the results depending on the chosen mechanism. That would

imply difficulties on choosing the aggregation mechanism to apply, since there is not a model that works in an optimal manner under all circumstances.

Furthermore, in highly dynamic environments, where the system conditions tend to change frequently, swapping the aggregation mechanism could be commonly required, which is not an easy task in current reputation management systems. After an analysis of the state of the art regarding solutions trying to address that issue, we found that some of the current works propose configurable or tunable reputation management models, at most. This solution may adjust some of the parameters defining the internal behavior of the given reputation model, but they do not provide enough flexibility to be applicable in dynamic environments.

In order to address that problem, we came up with a solution able to select and activate the most appropriate reputation management model on-the-fly, depending on the system conditions and desired performance measurements (Chapter 5). Hence, a number of reputation management models are available in our proposal, although only one of them remains active computing reputation values.

If the system detects that there is a model currently more appropriate to compute the reputation scores, which is determined from a number of pre-defined rules, such model becomes active. Besides, the solution incorporates mechanisms to allow a smooth transition between the current active model and the one to become active, preventing inconsistencies in the bootstrapping period required by the initialized model.

At the same time, we realized the lack of reputation management systems sensitive on preserving users' privacy. Within the context of user-centric identity management, reputation management systems attempt to gather users' recommendations about the services or other users (or even recommendations representing the trust that the services have amongst each other). These recommendations are considered private information, and freely distributing those recommendations opposes one of the goal that user-centric identity management systems should focus, that is, to protect users' privacy.

Therefore, after an analysis of the related work in that direction, and after studying the applicability of advanced cryptographic mechanisms, we proposed a method towards solving such shortcoming (Chapter 6). Using homomorphic encryption techniques, the proposed method allows computing recommendations provided by the users, yet preserving the privacy of those recommendations.

However, these techniques limit the application of some sophisticated mechanisms for aggregating recommendations, which count on detailed information about the users and the supplied recommendations. For instance, some reputation models compare users preferences, or their usage profiles, in order to provide customized reputation values. Nevertheless, that information is not available using the aforementioned techniques.

Hence, we decided to go one step further and, by defining a set of algorithms also based on homomorphic encryption, we proposed a system able to calculate customized reputation values, yet preserving the privacy of the users (Chapter 7).

By relying on an identity provider, the reputation management service is able to make use of the similarity between users, which is obtained by comparing the supplied recommendations between each other, to compute customized reputation values. Nevertheless, by applying the proposed techniques, neither the identity provider nor the reputation management service are able to determine the similarity between users, although customized reputation values are computed. Furthermore, they also cannot know the recommendations provided by a given user, therefore preserving their privacy.

III Results

The results of this thesis are essentially exposed in the articles that compose it. First, the results derived from analyzing the state of the art within the field of identity management systems have been presented in the book chapter entitled "Identity Management in Cloud Systems", published in the book Security, Privacy and Trust in Cloud Systems of the Springer editorial.

In this work, most common identity management standards and solutions have been described, identifying their benefits and drawbacks against a set of extracted requirements, which have been identified from typical use cases. Additionally, this work presents research challenges to be addressed in this context, as well as ongoing work, standardization activities and research projects aimed to address those challenges.

From such starting point, we investigate further into the common lack of trust management in current identity management systems, and the outcome was reflected in the article entitled "On the Application of Trust and Reputation Management and User-centric Techniques for Identity Management Systems", presented in the XII Spanish Meeting on Cryptology and Information Security (RECSI 2012). In this article, we study how reputation management systems can be combined with user-centric techniques within the context of identity management systems, setting the ground to address this important limitation.

Continuing this research line, we have defined a reputation management method and have described how it can be applied as an extension to the protocol defined by the OpenID standard. This work, entitled "Towards the integration of reputation management in OpenID", has been published in the Special Issue on Secure Mobility in Future Communication Systems under Standardization of the Computer Standards & Interfaces journal (Elsevier).

Such a reputation management solution is based on the idea that users can provide recommendations about a service, in such a way that those feedbacks can be aggregated by an OpenID Provider. The outcome of such an aggregation can be delivered to those users willing to use the service. Thus, users would be properly informed about the trust they can place in the service provider before actually interacting with it. Such an informed decision would increase the level of satisfaction of the users about the usage of the systems based on the user-centric identity management standard defined by OpenID. Furthermore, as part of the proposal, we present and analyze different mechanisms that the model may use to aggregate reputation values.

In order to analyze the proposed model, and each of the mechanisms to aggregate recommendations, a simulator implementation was conducted, whose description has been presented in the 19th European Symposium on Research in Computer Security (ESORICS), within the Security & Trust Management Workshop (STM 2014), with the name "ROMEO: ReputatiOn Model Enhancing OpenID Simulator". This simulator models diverse types of users and entities interacting between them, constituting different threats that the reputation management solution may be exposed to.

With this simulator some experiments were performed to analyze the behavior of the system, and to demonstrate that the solution is feasible, even when considering malicious entities or users. Additionally, one of the most interesting conclusions extracted from the analysis of the proposed model was to realize that there is not a unique method for aggregating the recommendations offering a better performance than the rest in all scenarios. Instead, the behavior of each aggregating mechanism rather depends on the current system conditions and expected performance measurements, which, in distributed environments, can become highly variable.

These findings are complemented with the outcomes of the article entitled "Dynamic and Flexible Selection of a Reputation Mechanism for Heterogeneous Environments", published in the Special Issue on Trustworthy Data Fusion and Mining in Internet of Things, of the Future Generation Computer Systems journal (Elsevier), and also described in the international patent entitled “System and method for determining a reputation mechanism” (see Appendix A.I). In such documents, we present a mechanism able to dynamically and automatically select the most suitable reputation model on-the-fly depending on the current system conditions and some desired performance measurements. The selection is based on a number of predefined rules using fuzzy sets, in order to ease the rule definition for administrators. These rules are set to represent the behavior of the reputation management models regarding each of the expected performance measurements, depending on the current system conditions.

Hence, it is no longer a matter of developing and deploying a configurable reputation model, but rather a pool of idle ones instead, and activate the most appropriate mechanism at each moment (chosen according to the defined rules), avoiding the need to stop or reconfigure the system. Moreover, the models swapping process is carried out in a smooth fashion, in order to prevent inconsistencies caused by the recently activated model, as it would need a bootstrapping period until initializing properly.

A set of experiments were conducted in order to validate the proposal, making use of the aforementioned simulator ROMEO. Thereof, it was proven that, with our dynamically exchangeable models solution, the computed reputation values are more accurate than those provided by traditional reputation management models, where a unique mechanism to compute the reputation is defined. Furthermore, the importance of performing the previously commented smooth transition was also analyzed.

Additionally, we propose a solution to address the lack of privacy that reputation systems described in the current literature present, taking eHealth scenarios as a reference due to their strong privacy requirements. Within the context of user-centric identity management, recommendations provided by users, as well as the opinion that the entities have among each other, are considered as private information and hence needs to be preserved. The outcomes of this research have been published with the title “Identity Management: In Privacy We Trust. Bridging the Trust Gap in eHealth Environments”, published in the Special Issue on Health IT Security and Privacy, of the IEEE Security & Privacy magazine.

In that manuscript, we propose a solution aimed to have the advantages of sharing such sensitive information in order to feed the reputation management systems, while keeping it private. By using homomorphic encryption, our approach shows how users’ recommendations and other required parameters to perform the reputation aggregation can be computed, yet not revealing those values to potential attackers.

Moreover, within the field of eHealth, we filed the patent entitled “Method to support an advanced home service coordination platform” (see Appendix A.II). In this case, we define an advanced emergency management scenario by combining identity management, access control and trust and reputation models. In this scenario, a reputation system assists a care coordinator to select the most appropriate care giver based on pacient’s needs, preferences, and other pacients’ recommendations. Only the selected care giver is able to access patient’s sensitive data, and only within the context of the corresponding care service, providing a more efficient care management while granting pacients’ privacy protection.

Finally, in the article entitled “Towards privacy-preserving reputation management for Hybrid Broadcast Broadband applications”, and accepted by the Elsevier Computers & Security journal, we present a solution defining a set of techniques also based on homomorphic encryption, which are able to compute customized reputation values, while preserving the privacy of the users.

This solution takes HbbTV (Hybrid Broadcast Broadband TV) as the baseline scenario, which is focused on providing an interactive TV, offering entertainment services on demand. The services may be offered via application stores, where users can rate and comment the available applications in order to advise future users.

The proposed solution makes use of an identity provider to hide the real identity of the users, although their recommendations and the similarity between users remain unknown to such an identity provider, as they are encrypted. Yet, despite the encryption, required calculation over these values to provide customized reputation values can be performed.

IV Conclusions and Future Work

User-centric identity management systems are of paramount importance to provide authentication, while preserving the privacy of the users, and enhancing interoperability between multiple domains. Those systems are designed as a solution to the Single Sign-On process, providing methods to share users' information between different entities.

By establishing trust relationships between different providers, the users are able to access different services making use of a unique identity, since a trustworthy entity is in charge of preserving their privacy. Nonetheless, user-centric identity management systems have shortcomings related to trust management in distributed environments, where establishing static agreements is no longer an option.

Reputation management systems have been applied in the last years as an alternative to handle trust in this kind of environments. Trust and reputation models gather recommendations from different sources attempting to predict the behavior of a given entity. However, the integration between reputation management systems and user-centric identity management systems is not straightforward, and several considerations and challenges have to be taken into account.

In this PhD thesis, a set of solutions has been proposed and analyzed in order to enhance user-centric identity management systems with reputation models, taking the peculiarities of distributed environments as a reference. In our opinion, this thesis can be considered as a guide assisting researchers willing to focus on this particular field.

It is interesting for any user to have mechanisms to find out the behavior of a given service provider before interacting with it. It is particularly significant when the service provider is requesting personal information to start the interaction. Due to the role that identity providers play within identity management systems, they seem to be the right candidates to supply the information about the service being accessed.

Notwithstanding the fact that identity providers are seen as trustworthy entities, in distributed environments we cannot rely on a static centralized provider to supply the reputation values of any service. Otherwise, this provider could become the goal of any attacker, or even provide biased reputation values for some colluding services.

Distributed environments raise specific challenges, concretely when private users' information come into play. Therefore, the application of reputation management systems needs to be analyzed appropriately, considering malicious users and entities which may collaborate between each other.

We have observed that it is usually cumbersome to find a reputation management model fitting suitably in any situation or behaving appropriately for any of the scenarios where to be deployed. Such circumstances lead to find numerous reputation management proposals, focused on concrete scenarios. It is therefore worth to work toward the unification of those models, analyzing their common elements, in such a way that only a small subset of internal components need to be swapped when a different reputation model wants to be applied. Such dynamically exchangeable reputation models solution makes system designers and administrators tasks easier, since they do not need to choose among several alternatives every time they need to deploy a reputation management model in a particular context.

Moreover, we should not forget that user-centric identity management systems should aim to protect the privacy of the users. This may seem to conflict with the functionality of reputation management systems, which try to collect as much users' information and recommendations as possible to provide accurate and customized reputation values. Nevertheless, as we present in this dissertation, researchers have already begun to find alternatives to address this dichotomy. By using certain cryptographic techniques, the benefits of gathering information could be leveraged when computing reputation values, without compromising the privacy of that information.

As future work, we propose to bring some of the mechanisms developed as part of this PhD thesis into a standardization body. That would ease the integration between reputation management systems and user-centric identity management systems in a near future. The idea behind would be to provide a set of best-practices, use cases and a recommendations guide to be followed by a designer of this kind of systems as a reference.

An appealing research line, also derived from this thesis' work, would consist on enhancing the proposed solutions in order to assist administrators on managing this kind of systems. Despite the proposal of a mechanism aimed to automatically select the most appropriate reputation management model at each moment according to some defined rules, such mechanism still would present some challenges for the administrators to define the mentioned rules.

In some scenarios, where the analysis of the intrinsic properties of the reputation model would be quite laborious, the idea would be defining a mechanism able to assist the administrators in that process. Furthermore, rule definition would be complemented with artificial intelligence techniques, in such a way that the system would be able to analyze the behavior of the different reputation management models, and adapt the rules accordingly.

Using advanced cryptogtaphic techniques to preserve users' privacy is definitely an attractive ongoing work. Used appropriately, these techniques may offer sophisticated reputation computation mechanisms, while hiding recommendations. The possibilities this field opens are immense, and its application as part of ongoing solutions is still a world to explore.

Regarding the integration between reputation management systems and user-centric identity management systems, in our opinion, we believe to have consolidated a milestone, but there still is much work ahead. Analyzing how this integration could be applied to other contexts, addressing requirements particular to other scenarios, where other challenges have to be faced, could constitute an interesting continuation of this PhD dissertation.

Resumen

I Motivación y Objetivos

Debido a la gran acogida de los sistemas de comunicación a través de Internet en los últimos años, sus usuarios intercambian cada vez más cantidad de información, entre la que se incluyen datos sensibles y personales. Pero estos usuarios, en ocasiones, no están suficientemente informados de cómo están siendo tratados sus datos personales, o desconocen quién podrá tener realmente acceso a éstos y con qué propósito.

Es muy habitual hoy día tener que rellenar formularios de registro incluso para servicios que probablemente se utilizarán una sola vez, como por ejemplo, añadir un comentario en un blog. Los datos requeridos suelen ser personales, como dirección de email o fecha de nacimiento, o incluso datos más privados que resultan innecesarios para la provisión del servicio en sí, como número de teléfono, aficiones, etc.

Esto no sólo implica tener que recordar distintos nombres de usuario y contraseñas para cada servicio, o exponernos a recibir correo no deseado, sino que también se pone en juego la privacidad de los usuarios. Cuando los usuarios proporcionan sus datos personales, en muchas ocasiones no conocen realmente cómo estos datos van a ser gestionados, a quién serán cedidos, o si, por ejemplo, podrán ser utilizados en campañas de marketing externas al servicio en el cual se están dando de alta.

Poseer información sobre estos usuarios se considera un activo cada vez más valioso, llegando incluso a convertirse en un objetivo para ciertas organizaciones con intereses comerciales. Esta información se utiliza especialmente con fines publicitarios, o con intención de desarrollar ataques avanzados sobre objetivos concretos en base a la información recopilada de dichos usuarios. Una gran cantidad de servicios ofrecidos a través de Internet intentan recopilar información de los usuarios, por ejemplo, a través de los anteriormente mencionados formularios de registro, o mediante otros medios, como intentar inferir perfiles de usuarios a través de las interacciones que éstos tienen con los servicios ofrecidos.

Tanto la privacidad, como el control sobre la información que otras entidades pueden obtener de uno mismo, son características cada día más reclamadas por los usuarios de cualquier sistema de comunicación. Además, estos temas están considerados en ciertos entornos geopolíticos como la Unión Europea, como un derecho de los usuarios. En este contexto se encuentran aquellos usuarios que no quieren relacionar su vida privada con las interacciones que realizan de manera diaria en los distintos sitios web, o aquellos que no quieren que se recopile información sobre sus preferencias y perfiles de uso. Por citar algunos ejemplos de este tipo de usuarios podemos encontrar, periodistas que quieren denunciar situaciones comprometedoras sin verse implicados en posibles represalias, militares que no pueden o deben revelar su situación geográfica, o simplemente como una medida más de seguridad para cualquier usuario de los sistemas de comunicación a través de Internet.

Una solución elegante para estos dilemas ha comenzado a extenderse recientemente mediante la utilización de sistemas de gestión de identidades. Los sistemas de gestión de identidades establecen relaciones de confianza entre distintas organizaciones, de manera que la información de los usuarios es gestionada por una entidad confiable, conocida normalmente como proveedor de identidad (por ejemplo, un ayuntamiento, universidad, proveedores de acceso a internet, proveedores de correo electrónico o cualquier otra organización conocida).

Cuando los usuarios acceden a un servicio en Internet (por ejemplo, una web de compra en línea, subscripción a un servicio de noticias, comentar una entrada de un blog, etc.), las funciones de autenticación y gestión de datos de los usuarios es delegada al proveedor de identidad. Así, este proveedor de identidad evita que los usuarios deban registrarse y envíen información privada directamente a los servicios que pretenden utilizar. Los servicios sólo obtienen la información que les envía el proveedor de identidad, el cual puede ocultar la verdadera identidad de los usuarios, protegiendo así su privacidad. Adicionalmente, los sistemas de gestión de identidades proporcionan la funcionalidad de Single Sign-On, esto es, permiten a los usuarios utilizar una única cuenta (la de su proveedor de identidad) para acceder a distintos servicios repetidamente. Es más, los sistemas de gestión de identidades están adoptando el paradigma “centrado en el usuario” (conocido como user-centric por su término en inglés), que adopta usabilidad y pone un mayor énfasis en las necesidades de los usuarios como factores determinantes.

Varias iniciativas han desarrollado diversas soluciones y estándares para definir la comunicación entre los proveedores de identidad y los servicios de Internet. Como ejemplos representativos podemos considerar SAML u OpenID. Sin embargo, estos sistemas, presentan numerosas carencias aún hoy día, especialmente en lo que se refiere al control que tiene el usuario sobre su propia información. Las soluciones actuales escasamente informan a los usuarios acerca de cómo es tratada su información personal, y raramente permiten a los usuarios decidir qué datos pueden ser o no cedidos a otras entidades.

Incluso, cuando los usuarios son informados de qué proveedor de servicios está solicitando su información, éstos desconocen cuánto pueden confiar en dicho proveedor. En otras palabras, los usuarios no saben cómo su información va a ser tratada, o si el proveedor de servicios con el que se pretende interactuar cumplirá con sus expectativas.

Estas dificultades se ven incrementadas cuando dichos sistemas de gestión de identidades han de ser desplegados en entornos distribuidos, como redes P2P, donde las relaciones de confianza no pueden ser gestionadas a través de servidores centrales estáticos. En estos casos, el establecimiento de contratos estáticos para fijar la calidad de servicio entre un proveedor de servicio y los clientes, como acuerdos de nivel de servicio (conocidos como SLA, por sus siglas en inglés), o para determinar cómo la información privada es gestionada, es de difícil aplicación. Esto hace necesario desplegar otro tipo de mecanismos para gestionar la confianza.

Como alternativa para la gestión de confianza en entornos distribuidos, los sistemas de gestión de reputación vienen implatándose exitosamente en los últimos años. En estos sistemas, la confianza que se puede depositar en una entidad dada (como por ejemplo, un nodo de la red, un proveedor de servicios o incluso un usuario) no viene establecida por contratos fijos, sino que está basada en experiencias pasadas que se han tenido con dicha entidad.

Los sistemas de gestión de reputación intentan predecir el comportamiento de una entidad a partir del comportamiento que ha tenido ésta en el pasado. La reputación se calcula normalmente a partir de la opinión que tienen otras entidades o usuarios que ya hayan interactuado con la entidad dada. De esta manera, cuando un usuario quiere interactuar con una entidad que no conoce, el usuario puede ser informado sobre el comportamiento que ha tenido dicha entidad en el pasado, y éste decidirá si continuar con la interacción o no.

Aunque se ha avanzado favorablemente tanto en la gestión de identidades como en la gestión de reputación, la integración de ambos conceptos no es directa en entornos distribuidos, y plantea una serie de retos que han de ser tenidos en cuenta. Por ejemplo, se han propuesto soluciones para dotar al sistema definido por el estándar OpenID de mecanismos de gestión de reputación. Sin embargo, esas soluciones están basadas en el establecimiento de servicios centrales fijos y confiables, que no encajarían en entornos distribuidos, en los que OpenID está basado precisamente.

En entornos distribuidos, no sólo no se puede contar con servicios centralizados confiables, sino que también permiten a cualquier usuario actuar con diferentes identidades y desplegar sus propios servicios. Si no se toman las medidas apropiadas, esto puede introducir ataques colaborativos, en los que se despliegan una gran cantidad de nodos con objetivos maliciosos, como por ejemplo aumentar o disminuir la reputación de otras entidades malintencionadamente.

Además, las condiciones del sistema pueden ser muy cambiantes en cuanto a la cantidad de usuarios, número de nodos desplegados, participación de los mismos, cantidad de usuarios maliciosos, etc. Esto requiere que los sistemas de gestión de reputación tengan que ser altamente dinámicos, de manera que se puedan adaptar a las variaciones del entorno.

Adicionalmente, para hacer funcionar los sistemas de gestión de reputación, los usuarios deben proporcionar recomendaciones sobre los servicios que han estado utilizando. De esta manera, las recomendaciones son agregadas utilizando diversos mecanismos, que incluso pueden producir valores de reputación personalizados a partir de las preferencias de los usuarios.

Sin embargo, eso podría implicar que el servicio encargado de agregar esas recomendaciones conociera las recomendaciones proporcionadas por cada uno de los usuarios, los servicios que han estado accediendo e incluso sus preferencias. Esto contradiría el fundamento de los sistemas de gestión de identidades centrados en el usuario, que deben buscar principalmente la protección de la privacidad de los usuarios.

Dados los retos que conlleva mejorar los sistemas de gestión de identidades centrados en el usuario mediante modelos de reputación en entornos distribuidos, en esta tesis se pretende:

- Estudiar el estado del arte de los sistemas de gestión de identidades y de los estándares definidos para los mismos, analizando las cuestiones abiertas, líneas de investigación y trabajo en curso que este tipo de sistemas plantea.

- Analizar el estado del arte en cuanto a los sistemas de gestión de reputación aplicables a escenarios de gestión de identidades.

- Identificar las ventajas y desventajas, así como las dificultades y retos que plantea la integración de sistemas de gestión de reputación con sistemas de gestión de identidades centrados en el usuario en entornos distribuidos.

- Diseñar y sugerir soluciones que permitan integrar sistemas de gestión de reputación en sistemas actuales de gestión de identidades centrados en el usuario en entornos distribuidos, de manera que los usuarios puedan determinar si un servicio cumplirá sus expectativas antes de interactuar con éste.

- Realizar un profundo análisis del comportamiento de dicha soluciones, utilizando distintos mecanismos para agregar las recomendaciones, y considerando usuarios y entidades maliciosas.

- Proponer y analizar soluciones para mejorar la capacidad de adaptación de los sistemas de gestión de reputación actuales en entornos dinámicos.

- Estudiar y plantear soluciones para aumentar la privacidad de los sistemas de gestión de reputación en el ámbito de los sistemas de gestión de identidades centrados en el usuario.

II Metodología

Esta tesis doctoral tuvo como punto de partida el análisis del estado del arte de los sistemas de gestión de identidades (Capítulo 1). Como parte de este análisis, identificamos aquellos requisitos esenciales que estos sistemas deben cumplir a partir del estudio de los casos de uso y escenarios más relevantes en el ámbito de la gestión de identidades centrada en el usuario.

Nuestra primera contribución consistía en la descripción de estándares de gestión de identidades representativos, junto con tecnologías y soluciones asociadas a los mismos, para destacar los beneficios y desventajas de cada uno de ellos con respecto a los requisitos identificados, haciendo hincapié en aspectos centrados en el usuario. Un resultado interesante mostró que no había una solución ideal que fuera capaz de cumplir todos los requisitos, enfatizando que aún quedan cuestiones importantes abiertas en este campo.

Uno de los resultados que más nos llamó la atención ponía de manifiesto la carencia de gestión de confianza que presentaban, en general, los sistemas de gestión de identidades en entornos distribuidos. Los sistemas de gestión de identidades analizados suponen la confianza entre las entidades, en algunos casos incluso regulada por acuerdos estáticos a nivel de servicio. Sin embargo, esta confianza es difícilmente aplicable en entornos más dinámicos.

Para solucionar esta carencia, nos centramos en una alternativa que viene implantándose exitosamente para gestionar la confianza en entornos distribuidos. Esto es, los sistemas de gestión de reputación. De esta manera, parecía razonable pensar que la gestión de la confianza en sistemas de gestión de identidades podía ser complementada mediante sistemas de gestión de reputación.

Así, continuamos realizando un profundo análisis de los sistemas de gestión de reputación, dentro del ámbito de los sistemas de gestión de identidades centrados en el usuario, y tomando como referencia las características intrínsecas de los entornos distribuidos. La idea era mejorar los sistemas de gestión de identidades para que éstos informen a los usuarios apropiadamente antes de utilizar los servicios a los que dichos usuarios quieren acceder, incluyendo entornos donde no es posible la gestión de confianza mediante contratos estáticos.

Sin embargo, pronto nos dimos cuenta de que la integración entre ambos mundos no era inmediata, debiendo tener en cuenta numerosas consideraciones para un correcto funcionamiento (Capítulo 2). La mayor parte de los trabajos existentes en este ámbito basaban la gestión de la reputación en servicios centralizados, siendo difícil su aplicación en entornos más dinámicos o distribuidos.

De este modo, decidimos diseñar un modelo de gestión de reputación (Capítulo 3), aplicado en un sistema de gestión de identidades centrado en el usuario, distribuido y ampliamente extendido actualmente, como es el caso de OpenID. Hasta donde pudimos comprobar, se trataba de una de las primeras soluciones que propusieran un sistema de reputación distribuido aplicado a dicho estándar.

Como parte del análisis de la solución anterior, investigamos varios mecanismos para agregar las recomendaciones recolectadas de los usuarios, diferenciándose en su complejidad, requisitos para hacerlos funcionar y en la capacidad para evitar usuarios malintencionados, entre otras cosas. De esta manera, no sólo analizamos si la solución propuesta es factible, sino también distintas formas de obtener los valores de reputación, cuyo comportamiento y precisión dependían de las condiciones del sistema (número de usuarios, participación de los mismos, porcentaje de usuarios maliciosos, etc.), así como de las medidas de rendimiento esperadas (recursos computacionales, consumo de recursos de red, precisión en los valores de reputación, etc.).

Para un correcto estudio de la solución anterior, llevamos a cabo la implementación de un simulador capaz de detallar el comportamiento del modelo de gestión de reputación propuesto frente a diferentes escenarios (Capítulo 4). Como parte de ese trabajo, describimos las amenazas a las que podrían estar expuestos los modelos de reputación dentro de ese ámbito. A partir de éstas, identificamos los elementos del escenario a ser modelados, incluyendo distintos tipos de usuarios y entidades malintencionados. Implementamos distintos mecanismos para agregar las recomendaciones, así como distintas gráficas para visualizar los resultados de la simulación desde varios puntos de vista. De esta manera, se podía estudiar el comportamiento de cada uno de los mecanismos del modelo de reputación descrito anteriormente frente a distintos escenarios.

Los resultados del análisis fueron positivos para demostrar la viabilidad del modelo. Sin embargo, el análisis de los distintos mecanismos para agregar recomendaciones puso de manifiesto la variabilidad de los resultados dependiendo del mecanismo escogido. Esto supondría dificultades a la hora de elegir cual es el mecanismo de agregación que debía ser utilizado en cada momento, ya que no había un modelo que funcionara de manera óptima bajo todas las circunstancias.

Es más, en entornos altamente dinámicos, donde las condiciones del sistema cambian constantemente, podría ser necesario estar reemplazando el método de agregación muy a menudo, que ciertamente no es tarea fácil en los sistemas actuales. Tras un análisis del estado del arte sobre soluciones tratando de remediar este problema, nos encontramos con que algunos trabajos actuales proponían, como mucho, modelos de gestión de reputación configurables. Estas soluciones podían calibrar algunos de sus parámetros que definían su comportamiento interno, pero esos medios no proporcionan la flexibilidad suficiente para ser aplicados en entornos dinámicos.

Para solventar ese problema, ideamos y analizamos una solución capaz de seleccionar y activar el modelo de reputación más apropiado sobre la marcha, dependiendo de las condiciones del sistema y de las medidas de rendimiento deseadas (Capítulo 5). De esta manera, se tienen varios modelos de gestión de reputación disponibles, aunque sólo uno de ellos permanece activo calculando los valores de reputación.

Si el sistema detecta, a partir de una serie de reglas definidas, que hay un modelo para obtener la reputación más apropiado que aquel que actualmente se encuentra en ejecución, dicho modelo pasará a estar activo. Además, la solución incorpora mecanismos para permitir que la transición entre el modelo activo y el que pasará a estar activo se lleve a cabo de manera suave, evitando posibles inconsistencias en el período de inicialización de los modelos.

Paralelamente, nos percatamos de la deficiencia de los sistemas de gestión de reputación a la hora de preservar la privacidad de los usuarios. Los sistemas de gestión de reputación, aplicados en el ámbito de la gestión de identidades centrada en el usuario, funcionan mediante la recopilación de las opiniones que los usuarios tienen sobre los servicios o sobre otros usuarios (o incluso la confianza que tienen los servicios entre sí). Esas opiniones son consideradas como información privada, y distribuir libremente las mismas contradice uno de los objetivos que los sistemas de gestión de identidades centrados en el usuario deberían priorizar, y que consiste precisamente en proteger la privacidad de los usuarios.

Así, tras un análisis del estado del arte en este sentido, y tras estudiar la aplicabilidad de mecanismos de criptografía avanzada, propusimos un método orientado a la solución de dicha carencia (Capítulo 6). Utilizando técnicas de encriptación homomórfica, el método propuesto permite el cómputo de las recomendaciones proporcionadas por los usuarios, pero preservando la privacidad de las mismas.

Sin embargo, estas técnicas limitan la aplicación de algunos mecanismos sofisticados para agregar las recomendaciones, que cuentan con información detallada sobre los usuarios y las recomendaciones proporcionadas. Por ejemplo, algunos modelos de reputación comparan las preferencias de los usuarios, o sus perfiles de uso, para proporcionar valores de reputación personalizados. Sin embargo, esta información no está disponible utilizando las técnicas mencionadas anteriormente.

Así que decidimos ir un paso más allá y, definiendo una serie de algoritmos basados también en encriptación homomórfica, propusimos un sistema capaz de realizar cálculos de valores de reputación personalizados, al tiempo que se preservaba la privacidad de los usuarios (Capítulo 7).

Apoyándose en un proveedor de identidad, el servicio de reputación es capaz de utilizar la similitud entre los usuarios, que es obtenida comparando las recomendaciones proporcionadas entre sí, para calcular valores de reputación personalizados. Sin embargo, con la aplicación de las técnicas propuestas, ni el proveedor de identidad, ni el servicio de reputación pueden determinar la similitud entre los usuarios, mientras que los valores de reputación personalizados pueden ser calculados. Es más, éstos tampoco pueden descubrir las recomendaciones proporcionadas por los distintos usuarios, preservando de esta manera su privacidad.

III Resultados

Los resultados de esta tesis han quedado esencialmente reflejados en los artículos que la componen. En primer lugar, los resultados del análisis del estado del arte de los sistemas de gestión de identidades se han presentado en el capítulo de libro titulado “Identity Management in Cloud Systems”, publicado en el libro Security, Privacy and Trust in Cloud Systems de la editorial Springer.

En este trabajo se han descrito los estándares y soluciones de gestión de identidades más comunes, identificando ventajas y limitaciones de los mismos, frente a una serie de requisitos extraídos a partir de casos de uso habituales. Adicionalmente, se presentan retos de investigación a abordar en este ámbito, así como trabajo en curso, actividades de estandarización y proyectos de investigación enfocados en dar respuesta a dichos retos.

A partir de ese punto, profundizamos más en la falta de gestión de confianza presente en los sistemas de gestión de identidades actuales, y los resultados se vieron reflejados en el artículo titulado “On the Application of Trust and Reputation Management and User-centric Techniques for Identity Management Systems” presentado en la XII Reunión Española sobre Criptografía y Seguridad de la Información (RECSI 2012). En este artículo, presentamos cómo los sistemas de gestión de reputación pueden ser combinados con técnicas centradas en el usuario, dentro del ámbito de los sistemas de gestión de identidades, para avanzar en la integración de estos dos ámbitos.

Continuando por esa línea, hemos definimos un modelo de gestión de reputación y hemos descrito cómo éste puede ser aplicado como una extensión del protocolo definido por el estándar OpenID. El resultado de este trabajo, titulado “Towards the integration of reputation management in OpenID”, ha sido publicado en el Special Issue on Secure Mobility in Future Communication Systems under Standardization, perteneciente al Computer Standards & Interfaces journal (Elsevier).

Dicho modelo de gestión de reputación está basado en la idea de que los usuarios puedan proporcionar recomendaciones sobre un servicio, de manera que éstas sean agregadas por un proveedor de identidad de OpenID. El resultado de tal agregación puede ser proporcionado a los usuarios que pretendan utilizar el servicio. De esta manera, los usuarios son informados sobre la confianza que pueden depositar en el servicio antes de utilizarlo. Esto aumenta el nivel de satisfacción de los usuarios con el uso de los sistemas basados en el estándar de gestión de identidades centrado en el usuario definido por OpenID. Adicionalmente, como parte de la propuesta, presentamos y analizamos distintos mecanismos que este modelo puede utilizar para agregar los valores de reputación.

Para analizar el modelo propuesto y cada uno de los mecanismos para agregar las recomendaciones, se llevó a cabo la implementación de un simulador, cuya descripción fue presentada en el 19th European Symposium on Research in Computer Security (ESORICS), dentro del Security & Trust Management Workshop (STM 2014), con el nombre “ROMEO: ReputatiOn Model Enhancing OpenID Simulator”. Este simulador modela distintos tipos de usuarios y entidades interactuando entre sí, configurando las distintas amenazas a las que la solución de gestión de reputación puede estar expuesta.

Con este simulador se llevaron a cabo experimentos para comprobar el comportamiento del modelo de gestión de reputación y demostrar que la solución propuesta es viable, incluso en presencia de entidades o usuarios maliciosos. Adicionalmente, una de las conclusiones más interesantes tras el análisis del modelo propuesto fue el hecho de que no existe un método para agregar las recomendaciones que ofreciera mejores resultados en todos los escenarios, sino que los resultados dependían de las condiciones actuales del sistema, así como de las medidas de rendimiento esperadas.

Estos resultados vienen complementados con los presentados en el artículo titulado “Dynamic and Flexible Selection of a Reputation Mechanism for Heterogeneous Environments”, publicado en el Special Issue on Trustworthy Data Fusion and Mining in Internet of Things, dentro del Future Generation Computer Systems journal (Elsevier), y descritos en la patente internacional titulada “System and method for determining a reputation mechanism” (véase el Apéndice A.I). En estos trabajos presentamos un mecanismo capaz de seleccionar dinámica y automáticamente el modelo de reputación más apropiado sobre la marcha, dependiendo de las condiciones actuales del sistema y las medidas de rendimiento deseadas. La selección está basada en reglas definidas utilizando conjuntos difusos que facilitan la definición de las mismas por parte de los administradores. Las reglas representan el comportamiento de los modelos de gestión de reputación a partir de las condiciones del sistema actuales y las medidas de rendimiento deseadas.

De esta manera, no se trata de desarrollar un modelo configurable, sino de tener varios modelos disponibles en espera, y activar el modelo más apropiado en cada momento (elegido según las reglas definidas), sin necesidad de detener o reconfigurar el sistema. Es más, definimos un mecanismo para que el cambio de modelo se haga de manera suave, para evitar posibles inconsistencias causadas por los modelos recién activados, ya que éstos necesitan un tiempo hasta inicializarse correctamente.

Un conjunto de experimentos se llevó a cabo para validar esta propuesta, utilizando el simulador mencionado previamente. A partir de los mismos demostró que los valores de reputación proporcionados son más precisos que aquellos obtenidos por modelos de gestión de reputación tradicionales, los cuales sólo definen un mecanismo para agregar las recomendaciones. También se analizó la importancia de realizar esa transición suave que comentábamos anteriormente.

Adicionalmente, abordamos el problema de la falta de privacidad que presentan los sistemas de gestión de reputación de la literatura actual, tomando escenarios de e-Health como referencia debido a sus fuertes requisitos de privacidad. En el ámbito de la gestión de identidades centrada en el usuario, las recomendaciones proporcionadas por los usuarios, o la opinión que tienen las entidades entre sí, son consideradas información privada, y por tanto necesitan ser protegidas. Los resultados de esta investigación fueron publicados bajo el título “Identity Management: In Privacy We Trust. Bridging the Trust Gap in eHealth Environments”, en el Special Issue on Health IT Security and Privacy del IEEE Security & Privacy magazine.

En ese artículo planteamos una solución que permite las ventajas que tiene compartir esa información para alimentar los sistemas de gestión de reputación, pero manteniéndola privada. Utilizando encriptación homomórfica, esta solución muestra cómo las recomendaciones de los usuarios y otros parámetros necesarios para realizar el cálculo de la reputación pueden ser calculados, sin necesidad de revelar dichos valores.

También, dentro del ámbito de la gestión de servicios médicos, registramos la patente titulada "Method to support an advanced home service coordination platform" (véase el Apéndice A.II). En este caso, definimos un escenario de gestión de emergencias avanzado combinando gestión de identidades, control de acceso y modelos de confianza y reputación. En este escenario, un sistema de reputación asiste a un coordinador de cuidados sanitarios a seleccionar el cuidador basado en las necesidades del paciente, preferencias y recomendaciones de otros pacientes. Sólo el cuidador seleccionado puede acceder a la información sensible del paciente, y sólo dentro del ámbito del servicio de cuidados correspondiente, proporcionando una gestión de servicios sanitarios más eficiente mientras que se garantiza la privacidad de los pacientes.

Finalmente, en el artículo titulado "Towards privacy-preserving reputation management for Hybrid Broadcast Broadband applications", que ha sido aceptado por el Computers & Security journal de Elsevier, presentamos una solución que define una serie de técnicas basadas también en encriptación homomórfica, capaz de realizar cálculos de reputación personalizados, preservando sin embargo la privacidad de los usuarios.

Esta solución toma HbbTV (Hybrid Broadcast Broadband TV) como escenario de referencia, el cual está orientado a proporcionar una televisión interactiva, ofreciendo servicios de entretenimiento bajo demanda. Los servicios pueden ser ofrecidos mediante tiendas de aplicaciones (app stores), en las que los usuarios valoran y comentan las distintas aplicaciones para aconsejar a futuros usuarios.

La solución propuesta utiliza un proveedor de identidad para ocultar la verdadera identidad de los usuarios, aunque las recomendaciones y la similitud entre los usuarios son desconocidas por dicho proveedor de identidad, al estar encriptadas. No obstante, dada la técnica especial de encriptación utilizada aquí, se pueden realizar los cálculos necesarios para proporcionar valores de reputación personalizados.

IV Conclusiones y Trabajo Futuro

Los sistemas de gestión de identidades centrados en el usuario son de una importancia primordial para proporcionar autenticación preservando la privacidad de los usuarios, a la par que mejoran la interoperabilidad entre múltiples dominios. Estos sistemas están diseñados como solución a los procesos de Single Sign-On, facilitando métodos para intercambiar información de los usuarios entre las distintas entidades.

A través del establecimiento de relaciones de confianza entre los distintos proveedores, los usuarios pueden acceder a diferentes servicios utilizando una sola identidad, y sería una entidad confiable quien se encargaría de preservar la privacidad de dichos usuarios. Sin embargo, los sistemas de gestión de identidades centrados en el usuario presentan ciertas deficiencias a la hora de gestionar la confianza en entornos distribuidos, donde el establecimiento de contratos estáticos no es una opción viable.

Los modelos de gestión de reputación han estado aplicándose en los últimos años como una alternativa para gestionar la confianza en ese tipo de entornos. Los modelos de gestión de reputación suelen recopilar recomendaciones de distintas fuentes para predecir el comportamiento de una entidad dada. Sin embargo, la integración de los sistemas de gestión de reputación con los sistemas de gestión de identidades centrados en el usuario no es inmediata, debiéndose tener en cuenta diversas consideraciones.

En esta tesis doctoral, se han propuesto y analizado un conjunto de soluciones para mejorar los sistemas de gestión de identidades centrados en el usuario con modelos de reputación, tomando las peculiaridades de los entornos distribuidos como referencia. En nuestra opinión, esta tesis puede ser utilizada como una guía que sirva de ayuda para los investigadores que pretendan centrarse en este campo.

Es interesante para cualquier usuario tener mecanismos para descubrir el comportamiento de un proveedor de servicios dado antes de interactuar con éste. Eso es especialmente significativo cuando el proveedor de servicios solicita información privada para comenzar la interacción. Debido al papel que juegan los proveedores de identidad dentro de los sistemas de gestión de identidades, éstos parecen los candidatos idóneos para proporcionar la información a los usuarios sobre los servicios que están siendo accedidos.

A pesar de que los proveedores de identidad son vistos como entidades confiables, en entornos distribuidos no podemos contar con un proveedor estático y centralizado para proporcionar los valores de reputación de cualquier servicio. En caso contrario, este proveedor se convertiría en el objetivo de cualquier atacante, o incluso éste podría proporcionar, de manera malintencionada, valores de reputación sesgados para algunos servicios.

Los entornos distribuidos presentan retos particulares, especialmente cuando la privacidad de los usuarios está en juego. Por tanto, la aplicación de los sistemas de gestión de reputación necesita ser analizada apropiadamente, introduciendo usuarios y entidades maliciosas, que pueden llegar a colaborar entre ellos.

Hemos visto que es difícil encontrar un único modelo de gestión de reputación adecuado para cualquier situación o apropiado para cualquier escenario donde ser desplegado. Esto nos lleva a encontrar numerosas propuestas de gestión de reputación, enfocadas en escenarios específicos. Por tanto, merece la pena trabajar hacia la unificación de estos modelos, analizado sus elementos comunes, de manera que sólo un subconjunto de componentes internos deba ser intercambiado cuando otro modelo de reputación quiera ser aplicado. Esto hace más fácil la tarea de administradores y diseñadores de sistemas, puesto que no necesitarían escoger entre las distintas alternativas disponibles cada vez que tuvieran que desplegar un modelo de gestión de reputación en un contexto determinado.

Como trabajo futuro, podría resultar interesante proponer algunos de los mecanismos desarrollados como parte de esta tesis en un cuerpo de estandarización. Esto facilitaría la integración de sistemas de gestión de reputación en sistemas de gestión de identidades en un futuro. La idea sería también proporcionar un conjunto de buenas prácticas, casos de uso y guía de recomendaciones a seguir, que los diseñadores de este tipo de sistemas pudieran tomar como referencia.

Otra línea de investigación interesante, también resultante de esta tesis, consistiría en mejorar las soluciones desarrolladas para ayudar a los administradores a gestionar este tipo de sistemas. Aunque se ha propuesto un mecanismo que selecciona automáticamente el modelo de gestión de reputación más apropiado en cada momento a partir de reglas definidas, esto aún podría suponer algún reto para los administradores a la hora de definir dichas reglas.

En ciertos escenarios, donde el análisis de las propiedades intrínsecas de cada modelo de reputación pudiera resultar muy laborioso, la idea sería definir algún mecanismo capaz de ayudar a los administradores en ese proceso. Además, la definición de reglas se podría complementar con técnicas de inteligencia artificial, de manera que el sistema fuera capaz de analizar el comportamiento de los diferentes modelos de gestión de reputación y adaptar las reglas en consecuencia.

El uso de técnicas de criptografía avanzada para proteger la privacidad de los usuarios es definitivamente una línea de trabajo futuro interesante. Utilizándolas apropiadamente, estas técnicas pueden ofrecer mecanismos de computación de reputación sofisticados, mientras que las recomendaciones permanecerían ocultas. Las posibilidades que abre este campo son inmensas, y su aplicación como parte de las soluciones en curso es aún un mundo por explorar.

Con respecto a la integración entre modelos de gestión de reputación con sistemas de gestión de identidades, en nuestra opinión, hemos consolidado un hito, pero aún queda mucho camino por recorrer. Analizar cómo esta integración podría ser aplicada a otros contextos, tomando requisitos particulares a otros escenarios, donde otros retos deben ser resueltos, puede suponer una interesante continuación de esta tesis doctoral.

Publications composing the PhD Thesis

1 Identity Management in Cloud Systems

Title:	Identity Management in Cloud Systems
Authors:	Ginés Dólera Tormo, Félix Gómez Mármol, Gregorio Martínez Pérez
Type:	Book chapter
Series:	Security, Privacy and Trust in Cloud Systems
Publisher:	Springer Berlin Heidelberg
Editors:	Surya Nepal and Mukkadim Pathan
Pages:	177-210
Year:	2014
ISBN:	978-3-642-38585-8
DOI:	`http://dx.doi.org/10.1007/978-3-642-38586-5_6`
State:	Published

Table 1: Identity Management in Cloud Systems

Identity Management in Cloud Systems

Ginés Dólera Tormo, Félix Gómez Mármol and Gregorio Martínez Pérez

1 Introduction

Identity management systems are of paramount importance to provide authentication and authorization based on end user identities trying to preserve privacy, while at the same time enhancing interoperability across multiple domains. Traditional identity management systems allow end users, to some extent, to manage their personal information for accessing certain services.

However, cloud computing brings a different perspective related to the end users' interests. New risks arise, especially due to the fact that the number of devices acting in the system grows exponentially [48]. In this regard, some kind of attacks could be more dangerous and the number of malwares to be considered and malicious users that could potentially join the system increases.

Additionally, end users are more concerned about how their data is managed, where it is located and who can access it. In this sense, cloud computing is changing some of the basic assumptions. As a result, any service in the cloud is exposed to trust, security and privacy threats that can compromise the identity of end users.

Improving the end users experience while offering certain novel identity-related features has been recently achieved by means of applying advanced identity management systems. These systems are designed to deal with authentication and authorization processes, enabling Single Sign-On and methods to exchange end users information between different entities and/or domains. By establishing trust links among

G. Dólera Tormo (✉) · F. Gómez Mármol
NEC Laboratories Europe, Kurfürsten-Anlage 36, 69115 Heidelberg, Germany
e-mail: gines.dolera@neclab.eu

F. Gómez Mármol
e-mail: felix.gomez-marmol@neclab.eu

G. Martínez Pérez
Departamento de Ingeniería de la Información y las Comunicaciones Facultad de Informática,
Universidad de Murcia, 30100 Murcia, Spain
e-mail: gregorio@um.es

S. Nepal and M. Pathan (eds.), *Security, Privacy and Trust in Cloud Systems*, 177
DOI: 10.1007/978-3-642-38586-5_6, © Springer-Verlag Berlin Heidelberg 2014

different providers, end users are granted to securely access different resources or services using a single identity, yet preserving end users privacy.

A lot of work has been done in this area, improving and adapting this kind of systems to different needs. A wide variety of identity management systems have been proposed to fulfill different requirements of particular environments or to deal with different challenges that certain contexts pose [18]. However, due to the variety of features offered by the different identity management systems, it is not trivial to determine which approach fits better in a given context.

Based on requirements and threats related to the cloud computing model, our main contribution is to present different identity management approaches, in order to analyze and compare how they fit in the cloud context. The questions these systems leave open and the ongoing work in this regard are described as well afterwards. We also provide a set of recommendations to be taken into consideration when designing or deploying any identity-based service in a cloud environment.

The remainder of the document is organized as follows. Section 2 presents some working groups, standardization activities and on-going international projects aimed to analyze identity-related challenges raised by current and forthcoming technologies. Section 3 presents some representative use cases on the field of identity management for cloud computing in order to help the reader to contextualize subsequent sections. Section 4 describes a set of requirements and threats to be taken into account when working with identity management systems. Section 5 presents a comprehensive survey on the most relevant identity management solutions existing nowadays applicable to cloud computing. Each approach is analyzed, showing its advantages and limitations. Finally, an extensive comparison of all these solutions is provided. Section 6 sketches some foreseen practical and realistic scenarios of application of advanced identity management techniques within the scope of cloud computing, while Sect. 7 extracts the current research challenges to be addressed in order to reach the aforementioned envisioned practical scenarios. To conclude, Sect. 8 presents some final remarks depicting the main findings of our research work.

2 Related Work

This section presents a set of related works in the field of identity management. It describes working groups, standardization activities and international projects whose objective is to identify, analyze, and describe identity management challenges and pending issues.

The OASIS Identity in the Cloud Technical Committee (OASIS IDCloud TC) [41] works to address the security challenges posed by identity management in the context of cloud computing. Its main goal is to collect and harmonize definitions, terminologies and vocabulary of Cloud Computing, and develop profiles of open standards for identity deployment, provisioning and management. Definition of protocols, APIs or implementations is out of scope of this TC.

OASIS IDCloud TC collects use cases to help identify gaps in existing Identity Management standards, and investigate the need for profiles to achieve interoperability within them, with a preference for widely interoperable and modular methods. Additionally, it works with standards bodies to recommend changes to existing standards trying to close current gaps. Use cases categories include identity infrastructure establishment, identity management, authentication, authorization, accountability and attribute management, security tokens, governance and audit and compliance.

The National Strategy for Trusted Identities in Cyberspace (NSTIC) [55] is a White House initiative to work with both public and private sectors in order to improve the privacy, security, and convenience of sensitive online transactions. It offers a collaborative vision to create the standards, policies, guidelines and recommendations needed for interoperable trusted credentials that would dramatically reduce identity theft and fraud online.

NSTIC introduces the so-called Identity Ecosystem, aimed at protecting end users privacy by helping them to verify that the websites they browse are legitimate, avoiding fake sites designed to steal personal information. Furthermore, it enforces service providers to follow a standard set of best practices in order to ensure end users that their personal data will be fairly handled, that they are informed on how their data will be used, and to enable them meaningful choices, while accountability features are deployed. For example, service providers would be required to collect and share the minimum amount of information necessary for authentication.

Additionally, the Kantara Initiative [36] is committed to help in driving policies and technical interoperability in order to verify trust in the identity-based experience of end users, Relying Parties and Federation Operators. Additionally, it works collaboratively to solve harmonization and interoperability challenges among identity-enabled enterprise, Web 2.0 and Web-based applications and services. The goal of this activity is to provide public and private sector organizations with uniform means of relying on digital credentials issued by a variety of identity providers in order to advance trusted identity and facilitate public access to online services and information.

Moreover, the Simple Cloud Identity Management (SCIM) specification suite [28], developed under the IETF, is designed to ease the use of identity management in cloud-based applications and services. SCIM seeks to build upon experience with existing schemas and deployments, placing specific emphasis on simplicity of development and integration, while applying existing authentication, authorization, and privacy models. Its intent is to reduce the cost and complexity of user management operations by providing a common user schema and extension model, as well as binding documents to provide patterns for exchanging this schema using standard protocols.

In turn, Identity Commons [33] is a community of groups working on developing the identity and social layer of the web. Its main purpose is to support, facilitate, and promote the creation of an open identity layer for the Internet, in such a way that control, convenience, and privacy for the individual are improved.

The main objective of the Web Identity Working Group [60], developed by the World Wide Web Consortium (W3C), is to provide Web developers with a secure and uniform access to elementary cryptographic operations, session state information, and authentication credentials for devices and applications like browsers. Web Identity Working Group aims to produce specifications that have wide deployment amongst end users, adopting, refining and extending existing practices and community-driven draft specifications.

Furthermore, Attribute-based Credentials for Trust (ABC4Trust) [1] is a research project funded by the 7th Research Framework Programme (FP7) of the European Union as part of the Trust and Security Program. The goal of ABC4Trust is to address the federation and interchangeability of technologies that support trustworthy yet privacy-preserving Attribute-based Credentials (ABC). ABC enhances classical trustworthy credentials, which normally do not respect privacy and reveal more information than required by the service. This project defines a common, unified architecture for ABC systems to allow comparing their respective features and combining them on common platforms, in order to deliver open reference implementations of selected ABC systems.

Likewise, PrimeLife [49] is another European Union project funded by its 7th Framework Programme. The main objective of the project is to bring sustainable privacy and identity management to future networks and services. This project addresses challenges related to end users digital interactions over the Internet, which involve leaving a life-long trail of personal data. PrimeLife advances the state of the art in the areas of human computer interfaces, configurable policy languages, web service federations, infrastructures and privacy-enhancing cryptography.

Finally, Secure Widespread Identities for Federated Telecommunications (SWIFT) [53] is also a European Union funded project within the 7th Framework Programme. The project leverages identity technology, building a cross-layer identity management framework as a key to integrate service and transport infrastructures for the benefit of end users and providers. It focuses on extending identity functions and federation to the network while addressing usability and privacy concerns for end users. SWIFT prepares the grounds for a new dimension of business dynamics allowing a fast entry of new players while expanding the business of existing ones.

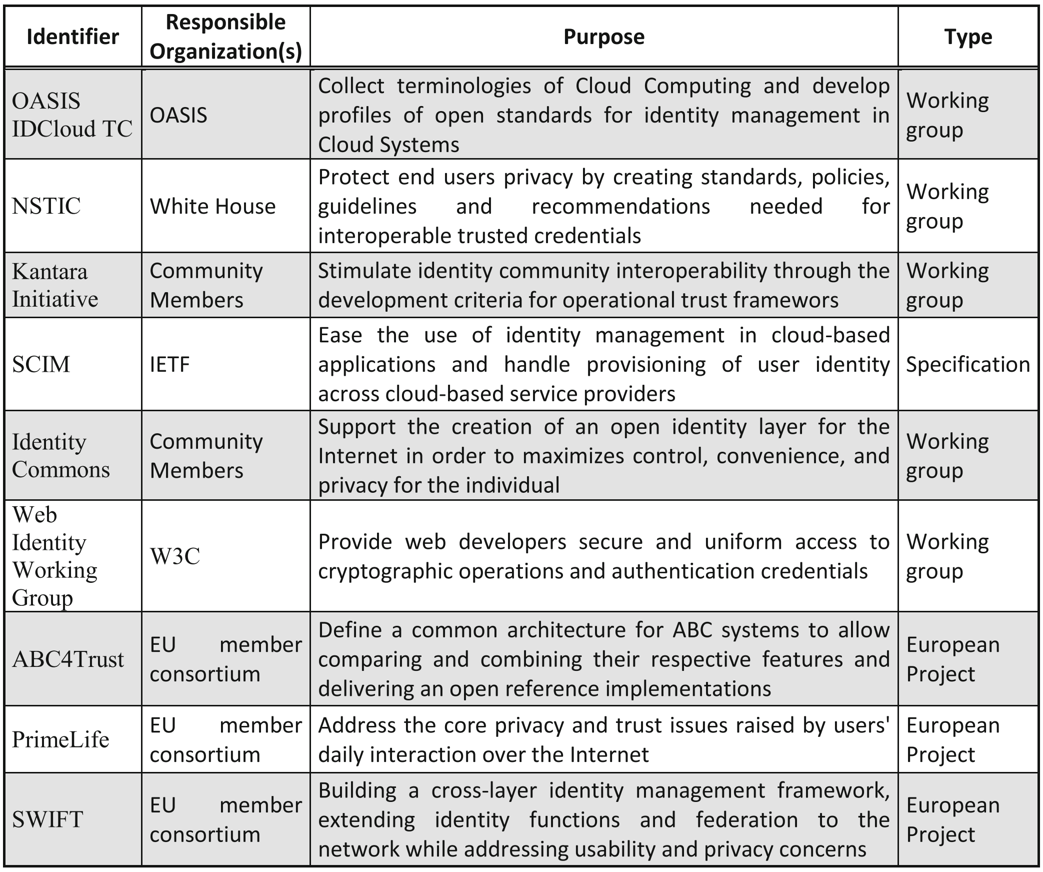
Table 1 summarizes the related work presented in this section.

3 Identity Management Usecases

This section will present some use cases on the field of identity management for cloud computing. In cloud environments, there could be several use cases identified regarding identity management, ranging from straightforward ones, such as those introducing common SSO concepts, to very complex scenarios, including for instance government provisioning or mobile customers.

The goal of this section is to present those use cases defining representative identity management scenarios which could help the reader to identify requirements, threats, features and challenges that these scenarios raise. This will establish the grounds to better understand the forthcoming sections. The use cases described here are inspired

Table 1 Overview of related work



Identifier	Responsible Organization(s)	Purpose	Type
OASIS IDCloud TC	OASIS	Collect terminologies of Cloud Computing and develop profiles of open standards for identity management in Cloud Systems	Working group
NSTIC	White House	Protect end users privacy by creating standards, policies, guidelines and recommendations needed for interoperable trusted credentials	Working group
Kantara Initiative	Community Members	Stimulate identity community interoperability through the development criteria for operational trust framewors	Working group
SCIM	IETF	Ease the use of identity management in cloud-based applications and handle provisioning of user identity across cloud-based service providers	Specification
Identity Commons	Community Members	Support the creation of an open identity layer for the Internet in order to maximizes control, convenience, and privacy for the individual	Working group
Web Identity Working Group	W3C	Provide web developers secure and uniform access to cryptographic operations and authentication credentials	Working group
ABC4Trust	EU member consortium	Define a common architecture for ABC systems to allow comparing and combining their respective features and delivering an open reference implementations	European Project
PrimeLife	EU member consortium	Address the core privacy and trust issues raised by users' daily interaction over the Internet	European Project
SWIFT	EU member consortium	Building a cross-layer identity management framework, extending identity functions and federation to the network while addressing usability and privacy concerns	European Project

on the Identity in the Cloud Use Cases working draft [51], which provides a set of use cases examining the requirements on identity management functions applied to the cloud.

3.1 UC01: Federated Single Sign-On and Attribute Sharing

There are numerous cloud services in the Internet offered by many different cloud service providers which, in turn, belong to many different domains. It is considered common for end users to have an account for each of these cloud services they want to use, having also different credentials for each of them. For example, they have to create a new account, protected by a specific username/password for accessing some resources or services offered by a given cloud service provider.

Federated Single Sign-On is a process that allows end users to access multiple services having a unique set of credentials. Once end users have been authenticated in their home domain, they do not need re-authentication for accessing different services, even if such services belong to external domains. Additionally, these services

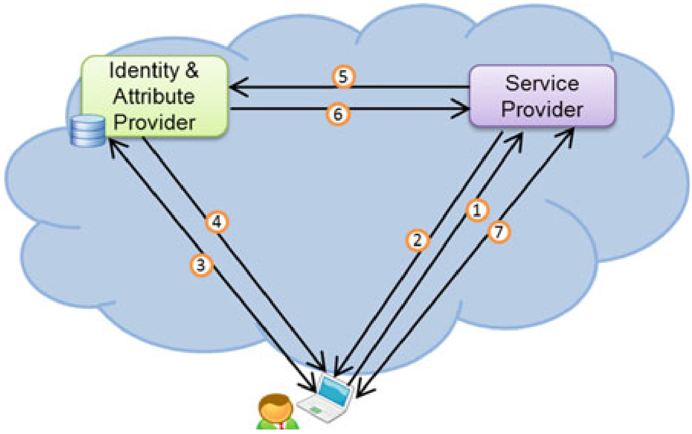


Fig. 1 Graphical representation and process flow of UC01

may also require retrieving users' information to provide the service or to perform some access control process.

Users' information is usually represented as attributes, such as age, country, postal address, etc. Again, to avoid users indicating such attributes for each service they want to use, identity management systems are planned in such a way that the service providers could recover the required attributes from their home domain, i.e., from their unique account, as long as a trust relationship exists between the querying domain and the domain providing such requested information. In this way, authentication and attributes could be asserted if they have been issued by a trusted party, although mechanisms to exchange such kind of information have to be defined (Fig. 1).

3.1.1 Process Flow

1. End user wants to access a service offered by a service provider
2. The service provider requires end users to be authenticated
3. End user is authenticated in her home domain
4. Home domain asserts user authentication
5. The service provider requires user attributes to offer the service
6. Home domain asserts user's attributes
7. End user accesses the service

3.2 UC02: Attribute Aggregation and Operations

End users usually have their attributes spread over multiple information sources, maintained by different providers in different domains. For example, academic information could be managed by their university, while information related to their postal address could be managed by their city hall and their credit card information is managed by their bank.

Although different information sources could mean different contexts, end users may want to present attributes maintained by different domains to the same cloud service provider at the same time. Furthermore, end users may want to perform some operations over the attributes in order to present just the required information to access a service. In this way, they could present some claims based on the attributes, but no the attributes themselves.

Service providers need to validate the received claims, for instance to check whether they (or the attributes they refer) are still valid. On the other hand, end users may want to present self-asserted attributes, describing some information about them, although such information is not asserted by a relying party (Fig. 2).

3.2.1 Process Flow

1. End user is asked by a service provider to present certain attributes, which could belong to different sources, in order to access a service
2. End user gathers attributes from different sources. Optionally, end user makes operations based on that attributes to claim the required information
3. End user sends the required attributes or the generated claims to the service provider
4. Service provider validates the attributes or claims
5. End user accesses the service

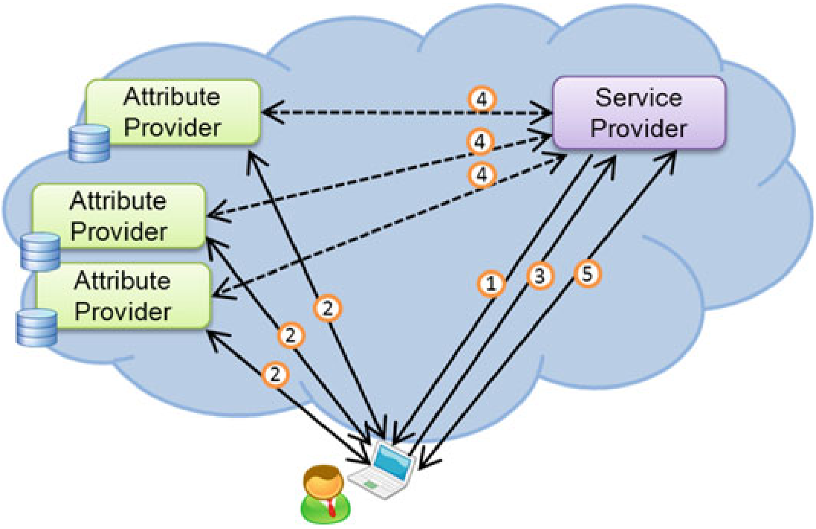


Fig. 2 Graphical representation and process flow of UC02

3.3 UC03: Identity Privacy in Shared Environment

Cloud service providers usually require end users attributes either to provide the cloud service itself (e.g. online shopping services require the postal address to send the purchased items), to perform access control (e.g. on-line film services may require the age of the end user to provide horror movies) or to provide customized services (e.g. a website showing different aspects according to the user language).

However, both end users identities and end users attributes are considered private information, and only reliable parties should gain access to them. In this way, end users identities must be hidden and they may give their explicit consent before any of their attributes is released. Furthermore, end users should release no more information than the strictly required by each cloud service provider, so they may want to choose which attributes will be released for each interaction with the cloud services.

To enable end users to achieve such a process, they do not only need the appropriate tools which allow selecting their attributes, but end users should be also properly informed about the service they want to interact, e.g. level of trustworthiness, fulfillment of privacy policies, etc. In this way, they can take the appropriate decision of allowing or not the service to obtain its attributes (Fig. 3).

3.3.1 Process Flow

1. A service provider requires end user attributes to provide a service
2. End user is informed that the service provider wants to access her attributes. In this step, detailed information of the service provider is shown to the end user
3. End user gives her explicit consent to release her attributes. Additionally, the end user selects the attributes which will be released

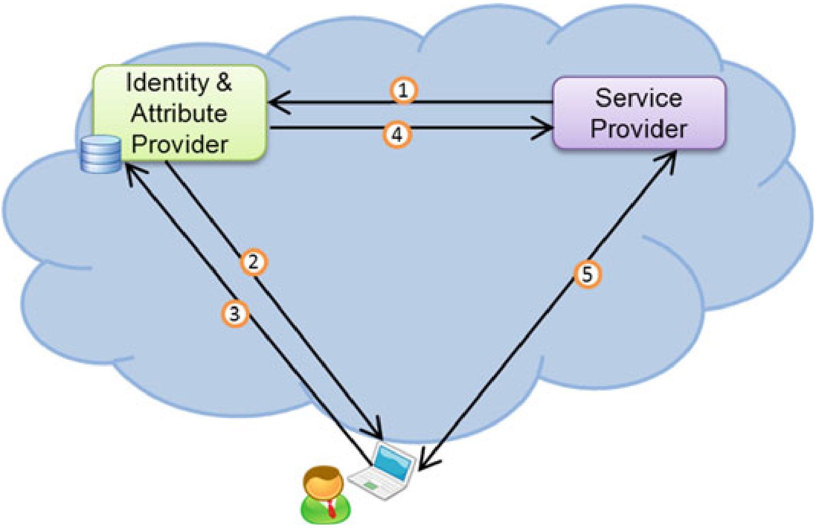


Fig. 3 Graphical representation and process flow of UC03

4. The service provider gets end user attributes
5. The service provider supplies the service based on the obtained attributes

4 Requirements and Threats

In a so demanding context, as is the case of cloud computing, identity management systems need to provide a set of features and give a minimum of guarantee that they properly fulfill the required behavior. Based on the described use cases, this section studies the main functional requirements as well as security threats to be considered when designing and deploying a new identity management system for cloud computing, or when selecting a currently existing one.

4.1 Requirements

Requirements have been grouped into three main categories according to their relation within the context of identity management systems, entitled (1) general requirements, which describes essential functionality that is expected from any identity-related system; (2) user-centric capabilities, which encompasses requirements related to the control offered to the end users for inferring in the interaction between different providers; and (3) information management functionalities, which defines the allowed operation that the end users have when they present information to a third party.

- General requirements:

  R1 **Confidentiality and integrity**: Since any identity management system makes use of sensitive information, they must assure that such information is shared only between appropriate entities. Additionally, they must guarantee that the information remains valid and complete even when is exchanged between different parties. In this way, identity management systems have to use secure communications channels and deploy the appropriate measures in order to ensure confidentiality and integrity of the information.

  R2 **Single Sign-On**: There are multiple services deployed in the Internet, belonging to many different domains, each of them managing their own set of credentials. A key requirement for the usability and security of any identity management system is to allow users of a domain to access applications hosted in another domain using the credentials of the domain they originally belong to. From the end users perspective, it is desirable to benefit from a SSO mechanism, avoiding having an account for each service they want to access.

  R3 **Logging and Auditing**: Logging and auditing discourage attackers since their attempt to subvert the system will be recorded and/or they will leave a trail which can be further tracked back to them. Moreover, if something unexpected happens, the lack of logs can result in significant difficulties while dealing with the occurred failure. Identity management systems must incorporate an effective logging and auditing mechanism able to trace relevant events happened in the system. This requirement guarantees that an end user or entities cannot deny having performed an operation or initiated a transaction, i.e., non-reputation is provided. To achieve this, the identity management systems may trace sent and received messages and audit (part of) their content, as well as internal operations.

R4 **Strong authentication**: Authentication mechanisms based on shared secrets such as common username-password authentication, do not offer enough protection avoiding impersonation or identity theft. In identity management systems, authentication mechanisms guaranteeing certain level of security need to be deployed, such as those based on biometric techniques or digital certificates, enhancing the level of security of the whole system.

R5 **Justifiable parties**: An identity management system must make its end users aware of the party with whom they are interacting while sharing information, and give certain indications about the level of reliability this party has. In turn, the relying party should also be able to confirm that the information presented by an end user is reliable, for instance if it has been validated by an authority.

- User-centric capabilities

R6 **End user consent**: When an identity provider needs to release some personal information about an end user, for instance when requested by a service provider to access a given service, the end user should be able to explicitly approve whether such information could be released or not. For example, the identity provider should show a confirmation page when some attributes are requested to permit the end users to decide if they want to continue, or not, with the transaction.

R7 **Control of accumulated data**: End users of identity management systems usually release some of their information to other entities, sometimes to enable another entity to manage their information on their behalf. Yet, the end users should be able to control which information each entity has about them, and to know how this information is being secured and protected. This is the case when the attributes of the end user are directly managed by the end user instead of by an identity provider.

R8 **Usability**: One of the main objectives of identity management systems is to ease any identity-related process to end users. This could not be achieved if end users are required to complete complex procedures, manage complicated tools or have advanced technical knowledge in order to interact with services. Instead, identity management systems should provide user-friendly interfaces and intuitive procedures when any identity-related functionality is presented to them.

R9 Off-line mode: Once an end user has authorized a transaction, the exchange of information between entities might be done without the intervention of the end user. Furthermore, if an attribute of the end user changes, the service provider should be able to get the updated value without needing interaction of the end user. For instance, a magazine service provider requires the postal address of their subscribed end users to send a printed version of their magazines and an end user makes use of her town hall identity provider to reveal her address. If the end user changes her address, it would be desirable that the magazine service provider automatically gets the new end user postal address without requiring the end user in such a process.

- Information management functionalities:

R10 Attribute aggregation: End users usually have multiple digital identities depending on the context they are involved. These identities could belong to different identity providers, each of them managing different kind of information. For instance, academic information of a given end user could be managed by the identity provider of her university, while information about her credit card is managed by the identity provider of her bank. Any user-centric identity management system should allow the end users to aggregate attributes from their different identities in order to present a combined set of claims to a given service provider at once.

R11 Attribute revocation: Some of the end users' attributes are not permanent but they can change throughout time or have an expiration time. Furthermore, the attributes could be revoked either by the identity provider which issued them or by the end user, for instance if she wants to cancel an account. When a service provider gets an end user's attribute, it should be able to check whether such attribute is still valid or not.

R12 Self-asserted attributes: End users' attributes usually need to be issued by an authority such as a trustworthy identity provider, in order to prove its validity. However, in some cases may be necessary to allow the end users to issue their own attributes if proving the validity of them is not mandatory. For instance, some service providers could require some non-critical attributes, such as hobbies, language or country, just to provide a customized service.

R13 Minimal disclosure information: The end users should be able to selectively reveal as less information as possible in the credentials presented to the service provider. For example, if an end user wants to make use of her driver license to prove she is older than 21, she should be able to extract a claim just related to her birth date without including the rest of information. Otherwise the service provider could gain all the other information contained in the driver license. Furthermore, the end users should be able to generate new valid claims based on others valid claims in order to prove that they fulfill a policy without revealing attributes. For instance, an end user should be able to prove that she is older than 21 making a verifiable claim based on her birth date but without actually revealing her birth date.

4.2 Threats

Identity management systems are exposed to a number of threats that can compromise its behavior when malicious users or entities try to subvert the system. We classify the threats into the next three categories:

- **Trust threats**: Identity management systems are aimed to simplify the end user experience by creating considerable trust complexity for both service providers and identity providers. They require an infrastructure where all the involved parties must be trusted for specific purposes depending on their role. However, if one of the parties acts maliciously, then the rest of the participants could be exposed to different risks. Identity management systems need to deploy mechanisms to allow entities to trust each other although in some scenarios the trust conditions could introduce some threats [17, 24, 25] if they have not been properly taken into account.
- **Security threats**: Any communication system is exposed to different risks which can compromise the security of the whole system. Malicious users are constantly coming up with new ways to attack any system, focusing their efforts on exploiting vulnerabilities of those systems. This is especially relevant for systems managing identity-related information, due to the fact that they potentially manage sensitive information [39]. Identity management systems need to avoid any threat which allows an attacker to affect negatively the system, from stealing information of the end users or acting on their behalf, to interfere in the communication or interrupt services.
- **Privacythreats**: Privacy is a desired feature of any communication system. End users usually want to keep the information of their digital identities secret. However, having information about the end users is increasingly being considered more and more valuable [38]. Furthermore, some organizations do not need to know the real identities of their end users, but they want to collect the behavior of each of them. Identity management systems have to deploy mechanisms to preserve end users' privacy. That includes (1) anonymity, where a service cannot know the real identity of an end user, (2) unlinkability, where a service provider cannot link different end user's accesses and (3) untraceability, where an identity provider cannot know the services that one of its end users has accessed.

5 Evaluation of Identity Management Approaches

In this section we introduce identity management standards, technologies and solutions which allow end users to manage their personal attributes required for accessing certain services. We analyze these approaches highlighting benefits and drawbacks of each in regards to the previously presented requirements. Finally, we summarize our analysis with a comparative table.

5.1 SAML

SAML [44], short for Security Assertion Markup Language, is an XML-based open standard which defines a framework for exchanging authentication, entitlement and attribute information between entities. In general terms, it allows entities, usually identity providers, to make assertions regarding the identity of end users to other entities, usually a service provider.

The first version of SAML (SAML 1.0) was adopted as an OASIS standard in November 2002. Since then, a minor revision (SAML 1.1) was made in November 2003, and a major revision (SAML 2.0) was made in March 2005, which is the most widespread version. Several organizations are offering support to SAML 2.0, being Shibboleth [20] the reference solution for this standard.

In the common workflow of SAML, as shown in Fig. 4, an end user wants to access a service from a service provider, but this service provider needs to authenticate the end user and obtain some attributes about her. The authentication process, instead of being performed by the service provider, is delegated to the identity provider, which is in charge of managing the user's identity.

To this end, the service provider redirects the end user to her identity provider along with a SAML Authentication Request. The identity provider asks the end user for her credentials, for instance with the usual username/password form, although other authentication methods could be used. Once the identity provider validates the authentication, it redirects the end user back to the service provider along with a SAML Assertion indicating that the end user has been authenticated.

At this point, the service provider may request some attributes of the end user sending SAML Attribute Query messages directly to the identity provider. Since the end users do not have to manage their attributes but they are managed by the identity provider, this solution is usually easy to use and no technical knowledge is required from the end users.

Nevertheless, end users are not aware of which attributes are being released. In fact, it is hard for them to control the information that a given identity provider accumulates about them. Even though some implementations allow defining some

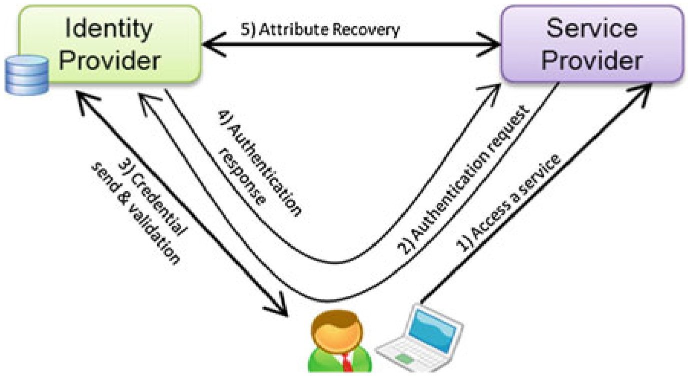

Identity Provider
5) Attribute Recovery
Service Provider
4) Authentication response
3) Credential send & validation
2) Authentication request
1) Access a service

Fig. 4 SAML general workflow

attribute release policies, they are also managed by the identity provider. Additionally, end users are not asked for consent before releasing their attributes.

Each identity provider is also in charge of managing the trust relationships with the service providers. The assertions indicating authentication and attributes are digitally signed by the identity provider, in such a way that their validity could be verified by the service provider. However, there is no option for the end users to present self-asserted attributes to a service provider. Furthermore, even though some SAML identity providers allow obtaining attributes from different information storages, an end user could not present assertions from different identity providers at the same time.

The issued assertions make use of pseudonyms preserving the end user's privacy. That is, the service providers do not know the end user's real identity. However, the identity provider could trace all end users accesses since it has to generate an assertion each time they need to access a service provider. Additionally, end users should initialize the transaction to allow the service provider get attributes, making the offline mode requirement hard to achieve.

One of the main purposes of SAML is providing a SSO mechanism for accessing different service providers with a unique account [15]. Hence, if the unique password of an end user is stolen, the thief could gain access to such services providers on behalf of the end user. Furthermore, a malicious service provider could redirect the end users to a fake identity provider, presenting a similar aspect to the original one, asking for inserting username and password, in order to steal passwords if they do not realize it is a malicious website (the so called "*phishing attack*" [34]).

5.2 OpenID

OpenID [50] is an open technology standard which defines a decentralized authentication protocol in order to allow end users to sign in to multiple websites with the same account. The original OpenID authentication protocol was developed in May 2005, and its current 2.0 version is maintained by the OpenID community since 2007.

With over one billion OpenID enabled end user accounts and over 50,000 websites accepting OpenID, this standard is nowadays widespread in the Internet. It is supported from several large organizations such as AOL, Google, Microsoft, VeriSign and Yahoo!, among many others [46]. Yet, OpenID is not owned by anyone, but the OpenID Foundation in in charge of assisting the community.

When end users create an account in an OpenID provider (i.e. identity provider) they receive an identifier as a URL or XRI. Then, when they access a website which requires authentication and supports OpenID (i.e. relying party or service provider) they may enter their identifier in order to be redirected to their OpenID provider, as shown in Fig. 5. It is worth mentioning that this identifier is usually unique for each end user (e.g. alice.myopenid.net). Therefore, the service provider could trace the end user accesses, since they always use the same identifier.

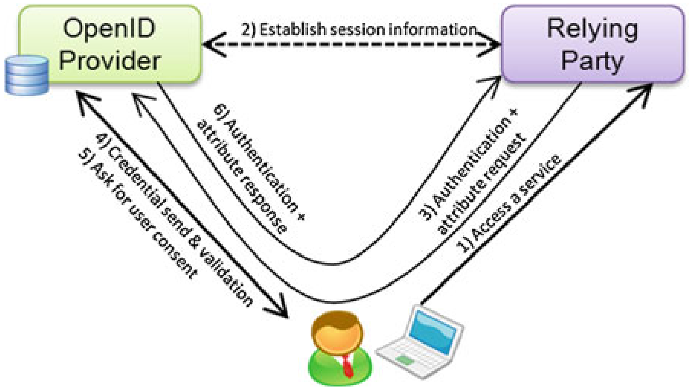

OpenID Provider
Relying Party
2) Establish session information
1) Access a service
3) Authentication + attribute request
4) Credential send & validation
5) Ask for user consent
6) Authentication + attribute response

Fig. 5 OpenID general workflow

Being in the OpenID provider, end users could make use of their unique username/password to perform the authentication process. After checking the end user's credentials, the OpenID provider shows a confirmation page, where the end user could verify and select the information which will be released to the service provider. That is to say, end users can check or uncheck the attributes to be released; yet, they cannot make claims based on such attributes.

Finally, end users are redirected back to the service provider, which now can have the requested information about the end users, but neither their password nor their real identity. As commented, the end users' information is managed by the OpenID provider, which makes the system easy to use although requires the user to be online to perform a transaction. Additionally the users can hardly control the information that the OpenID provider gain about them. Furthermore, the OpenID provider could trace end users' accesses since it establishes direct communication with the service provider each time the end user needs to provide an authentication assertion or release some attributes [30].

Even though the attributes are issued by an OpenID provider, the service provider cannot determine the reliability of such an OpenID provider. Since the framework is aimed to distributed environments, where no Certification Authority should be implied, any entity could become an OpenID provider, just implementing the OpenID protocol [26]. The end users can therefore present self-asserted attributes whose validity does not have to be validated by the service provider. However, end users cannot aggregate attributes from different OpenID providers.

The OpenID protocol also presents some security issues [16]. Similarly to the previous standard, if the password of an end user is stolen, the thief could access all the services accepting OpenID on behalf of the end user. Furthermore, a malicious service provider could redirect end users to a fake OpenID provider in the authentication process simulating the real end users' OpenID provider to steal their password.

5.3 OAuth

OAuth [29] is an IETF specification which defines a protocol in order for clients to access server resources on behalf of a resource owner. It provides a process for end users to authorize third-party accesses to their server resources without sharing their credentials. The first specification of the OAuth protocol dates from December 2007, although that version was updated on June 2009 to address the session fixation attack [37] and published as RFC 5849 [27] in April 2010.

Currently, there is a working draft in progress of OAuth 2.0 which is being supported by several companies, such as Facebook, Google and Microsoft. OAuth 2.0 is not backward compatible with OAuth 1.0, although the latter is also extendedly supported by several services providers such as LinkedIn, MySpace, Twitter and Yahoo! [45].

In the common workflow of the protocol, as depicted in Fig. 6, an end user wants to share some of their private resources which are maintained in a server (i.e. identity provider), like photos or documents, with a client (i.e. service provider) without revealing any password to such client.

To achieve this process, end users access the client website, which requests the end users' resources. Since the client supports OAuth, the end users may select the identity provider where the resources will be obtained from. At this point, the client requests a set of temporary credentials from the identity provider, and once received the end user is redirected to the identity provider with those temporary credentials.

End users can now see their identity provider website, where they could perform the common username/password authentication process without revealing their credentials to the client. After performing the authentication process, the end user is asked to grant (or deny) access to the client for getting some of their resources.

Despite its several advantages, this solution does not allow either attribute aggregation between different sources, nor making claims based on attributes, but just releasing them. In fact, the end users cannot choose which attributes will be released, just permit or deny the access to a set of them. Furthermore, the granularity of the set

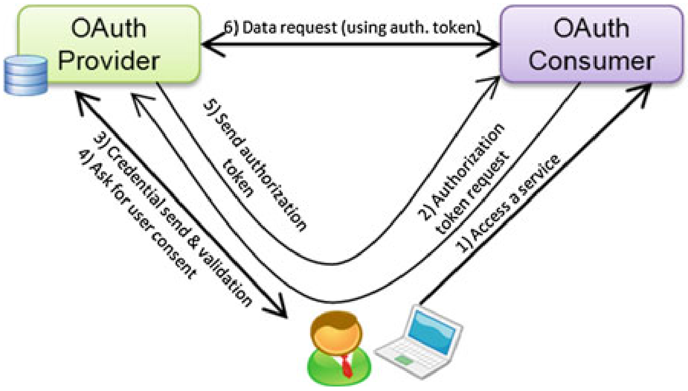

OAuth Provider
6) Data request (using auth. token)
OAuth Consumer
5) Send authorization token
2) Authorization token request
1) Access a service
3) Credential send & validation
4) Ask for user consent

Fig. 6 OAuth general workflow

of attributes or resources to be released depends on the OAuth server. For example, an end user might need to share a whole photo album even if she just wants to share one picture from it.

If the end users approve the request, they are redirected back to the client website, indicating the temporary credentials identifier, informing that they have completed the authorization process. Then, using its temporary credentials, the client requests a set of token credentials to the OAuth server, which will be used for requesting the resources.

Once the client has the set of token credentials, the communication is directly done with the OAuth server without requiring the user to be online. Using this process, the OAuth server hides the real identity of the end user to the client. However, the OAuth server can trace end user's accesses since it communicates with all the services the user wants to make use of.

By establishing a direct communication between OAuth server and client, this solution lets the OAuth server revoke attributes if they are not valid any more, but on the other hand it is difficult for the users to invalidate the granted authorization if they do not want it any more. At most, users could establish some expire period when they confirm the authorization.

Additionally, the client does not know how much can trust in an OAuth server and the validity of the provided attributes. This protocol makes no attempt to verify the authenticity of the server. This solution is mainly focused on allowing access to resources, where the validity of the resources does not have to be validated by an authority. In this sense, this solution allows self-asserted attributes, although it depends on the functionality offered by the OAuth server.

Similar to the previous solutions, this approach is easy to use, since it just shows some user-friendly web pages, although controlling the accumulated data is hard to achieve since it is managed by the OAuth server. It also has similar security problems regarding stolen passwords, since the same account is used for accessing different services. In the same line a malicious client could redirect the end users to a fake OAuth server in the authentication process trying to steal their passwords. Additionally, OAuth 2.0 tokens are not signed, which tends to simplify the protocol although it must rely on SSL/TLS channels to establish a secure communication, making this protocol vulnerable to Man-in-the-middle attacks [8].

5.4 Cardspace

Windows CardSpace [13, 40], also known as its codename InfoCard, is the Microsoft client or Identity Selector for the Identity Metasystem [35]. Although in February 2011 Microsoft decided not to ship this project any more, it is worth describing this solution since it provides the basis for future technologies such as U-Prove [47].

Taking into consideration that the end users may have different identities depending on the context where they are interacting, the challenge of this approach is to allow the end users to create, use, and manage their diverse digital identities in an understandable and effective way. For instance, at work end users might have a digital identity assigned by their employer while they maintain a private digital identity in MySpace.com to share some music content.

The idea behind Windows CardSpace is that end users could manage their digital identities, and their related attributes, in a similar way that they manage their cards in their wallets. In this sense, when end users are requested to present some information about them, they open their Identity Selector (their wallet), select one of their cards which contains the requested information, and present it to the requester. The Information Cards are usually issued by trustworthy entities, in order for the requester to verify the validity of the information contained.

To achieve such a process, as summarized in Fig. 7, the requester (i.e. relying party or service provider) supplies some requirements to the end user, such as which claims it expects. The Identity Selector shows the cards which fit the requirements to the end user. Once the end users select a card, the Identity Selector requests to the issuer of this card (identity provider) a security token containing the needed claims. Security tokens are signed to check the identity of the identity provider, establishing this way trust relationships. Finally, the Identity Selector sends the security token to the requester, which could be validated making use of the signature.

Windows CardSpace is entirely agnostic about the format of the security token [4]. For this reason, CardSpace can work with any digital identity system, using any type of security token, including X.509 certificates, Kerberos tickets, SAML assertions, etc.

In contrast to the previous solutions, the information of end users is managed by the end users themselves. The user could see all the information contained in an Information card which eases controlling the accumulated data. Furthermore, since the end user selects the information card to be sent, there is an explicit user consent, which also means that the interaction with the user is needed.

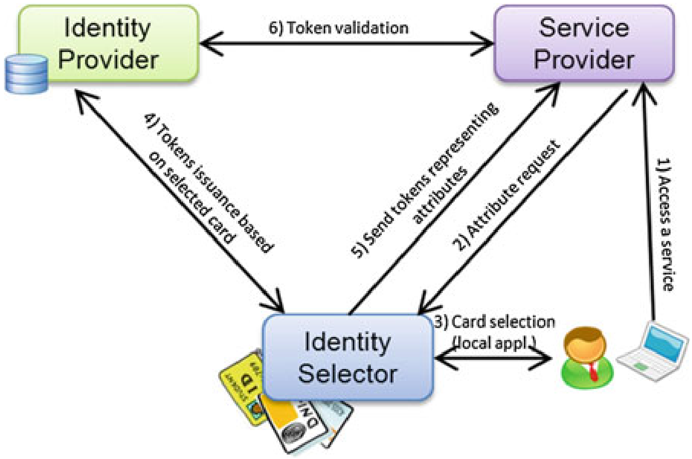

Identity Provider
6) Token validation
Service Provider
4) Tokens issuance based on selected card
5) Send tokens representing attributes
2) Attribute request
1) Access a service
3) Card selection (local appl.)
Identity Selector

Fig. 7 CardSpace general workflow

However, end users need to manually hold their cards, for instance by installing the cards in their devices. Hence, if the users move to another device they need to carry their cards with them in order to install such cards in the new device in use. This presents further difficulties if the end users want to make use of their cards in a public computer. Additionally, the implementation of reference is integrated with Microsoft Windows, which makes hard to be used in others operating systems, or devices, such as mobile phones.

Although Cardspace could make recommendations about the card to make use of, it is the user who needs to check the information to be released and decide how trustworthy the service provider is. In addition to that, the end user can neither select which attributes inside a card will be released, nor aggregate attributes from different cards, nor make claims based on attributes. The whole card is presented instead.

This solution allows the end user to create personal cards, as if they were issued by a self-identity provider. They usually include personal information such as name, addresses or phone numbers, which does not have to be verified. However, in this case a service provider could trace users' accesses, since the self-issued identity provider uses the same private key for each service provider to sign the tokens.

Moreover, the identity provider generates the user claims each time the user accesses a service in order to form the security tokens. In this way, the identity provider could not directly know the service the user is accessing, but it could trace which identity cards are being used and when. Furthermore, even though by default the service provider's identity is not revealed by the Identity Selector to the identity provider, when a token is requested, the identity provider might require knowledge of the service provider's identity before issuing the requested token (e.g. for creating a Kerberos ticket for that specific service) [2].

When some attributes need to be revoked, the identity provider just has to stop honoring request for security tokens made with this card. If the users want to revoke a card (e.g. from a stolen laptop) they should contact the identity provider. However, self-issued cards could not be revoked, and users should contact each service provider asking for not accepting the self-issued cards.

In order to avoid impersonation, the username/password mechanism for authentication is replaced by using the information card. However, how to acquire information cards is not defined, but it depends on the identity provider. In this sense, if the identity provider does not provide the appropriate measures, the information cards could be stolen and the user could be impersonated [21].

On the other hand, the Information card does not have to contain sensitive data, such as credit card number, but it is maintained in the identity provider, and is released in the security token instead. Additionally, CardSpace improves how websites prove their identity to the users by introducing higher-assurance certificates. These certificates also enable a way for those users to learn the level of assurance a site is offering, which could help them to take decisions about whether to trust a given website.

5.5 Higgins

Higgins is an open source identity framework designed to enhance the end user experience, by integrating identity profiles and social relationships information across multiple sites. The firsts versions of Higgins [19] (Higgins 1.0 and Higgins 1.1), released on 2008, offer an Identity Selector service for the Identity Metasystem. The latest version, Higgins 2.0, is still under development and is planning to implement a Personal Data Service (PDS).

A PDS is a cloud-based service that works on behalf of the end users, giving them a central point for controlling their personal information. It not only manages end users' attributes, but it also manages data flows to external businesses and to other end users' PDS.

As shown in Fig. 8, the functionality and workflow of Higgins is similar to Microsoft CardSpace, but in contrast to it, the cards in Higgins are maintained by a hosted service, outside the devices of the end users. Hence, the Identity Selector of Higgins is mainly a thin client that only implements the end user interface, while the core functionality is performed by the hosted service. In this way, the end users could make use of different devices to access their cards without having to carry them.

However, the end users need a specific piece of software installed in their device as a client application, such as a plugin in the web browser, to make use of the Identity Selector. There are implementations for different operating systems and some intuitive card selectors available, but since the end users need to directly manage their cards, using this solution is not trivial for inexpert users.

Higgins introduces a new kind of Information Card, namely the relationship cards (r-cards) [56]. These cards allow an end user to establish a data sharing relationship between an identity provider and a relying party. In this way, the relying party could directly request end user's attributes to an identity provider without interacting with the end user, but the end users control the authorization to their data.

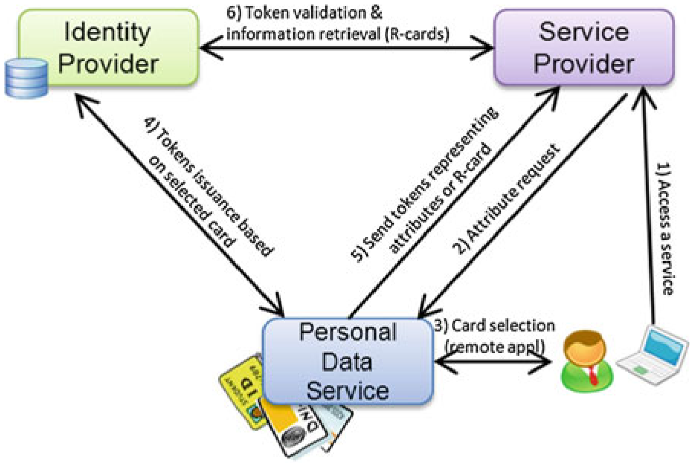

Identity Provider
Service Provider
6) Token validation & information retrieval (R-cards)
4) Tokens issuance based on selected card
5) Send tokens representing attributes or R-card
2) Attribute request
1) Access a service
Personal Data Service
3) Card selection (remote appl)

Fig. 8 Higgins general workflow

Either issuing specific credentials for a service provider as CardSpace does, or interacting directly with the service provider through a relationship card, the identity provider could trace users' accesses. Furthermore, the hosted Identity Selector should be placed in a server that the end users really trust, since it not only can trace all their interactions with the different relying parties, but also it is in charge of managing and storing all their identities information.

Regarding user consent, minimal information disclosure, attribute aggregation and revocation requirements, this solution presents similar issues to the CardSpace solution previously presented, since both are based on Information cards.

5.6 U-Prove

U-Prove [47, 57] is a cryptographic technology which presents a type of credential or token to encode end users' attributes in such a way that the issuance and the presentation of such tokens remains unlinkable. U-Prove was developed by Credentica in 2006, acquired and maintained by Microsoft since 2008.

The U-Prove technology makes use of Zero-knowledge proof methods [22, 23] to issue the tokens. Zero-knowledge proof is a way for an end user to prove possession of a certain piece of information without revealing it. That is, an end user can provide an assertion containing a set of attributes revealing nothing beyond the validity of such an assertion and the attributes. In this sense, this method offers the same level of security as X.509 certificates, with additional privacy protecting features.

In a similar way than end users manage Information cards, they may manage U-Prove tokens, as depicted in Fig. 9. These tokens are obtained from different identity providers, which prove the validity of such tokens. Therefore, when a service provider requires some end users' attributes, the end users could present one of their tokens, with the peculiarity that the identity provider is not involved in this process.

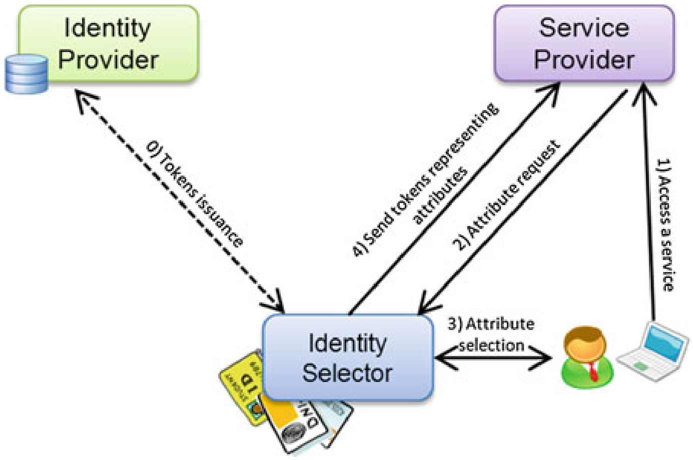


Fig. 9 U-Prove general workflow

That is, the identity provider and the service provider do not have to establish any communication between them.

Furthermore, even if the service provider is the same entity than the identity provider, the identity of the end user presenting the token could not be revealed, due to the fact that the token does not provide information regarding the issuance process which could be traced.

In this solution, the users control their information by themselves, which raises some disadvantages in terms of usability as previously commented when presenting other solutions (i.e. CardSpace and Higgins). On the other hand, end users could decide which of the attributes contained in a token will be released, without presenting the whole token, achieving part of the minimal disclosure information requirement.

However, it does not allow making claims about an attribute without revealing the attribute, for instance in order to prove that the value of an attribute is within a certain range. Although some mechanisms have been proposed to solve this issue [6], they are not efficient. In a similar way, combining tokens or attributes issued by different identity providers, in order to present a set of aggregated attributes at the same time to a relying party, is not available in this solution.

Although U-Prove tokens do not reveal the real identity of the end users, the service provider could trace different accesses of a given end user due to the fact that her tokens present the same public key and signature. This could be solved if the identity provider issues many different tokens to the end user with the same attributes [47], but this solution could be impracticable for large scenarios.

Attribute revocation is available for the users if they contact the service provider to invalidate one of their attributes. Nevertheless, revocation from the identity provider side is hard to achieve and it is an open research question for this technology. Due to the fact that the identity providers are not involved when the end users present a token, the user could be presenting a token even though it has been already revoked by the identity provider. Ordinary Certificates Revocation Lists (CRL) cannot be used since they would break the unlinkability capability of this solution. Some solutions based on unlinkable blacklists have been proposed [7] although they are not practical for large blacklists.

5.7 Idemix

Idemix, short for Identity Mixer [32], is an anonymous credential system following the protocols described by Camenisch and Lysyanskaya [9] in order to allow the end users to control the dissemination of personal information and preserving their privacy. Idemix has been developed at IBM Research and the first design and implementation document [10] dates from 2002.

Idemix makes use of Zero-knowledge proof methods to generate credentials. Similar to U-Prove, an end user can obtain credentials containing attested attributes from identity providers, and prove to a service provider the validity of such attributes without revealing any other information. The credentials are maintained by the end

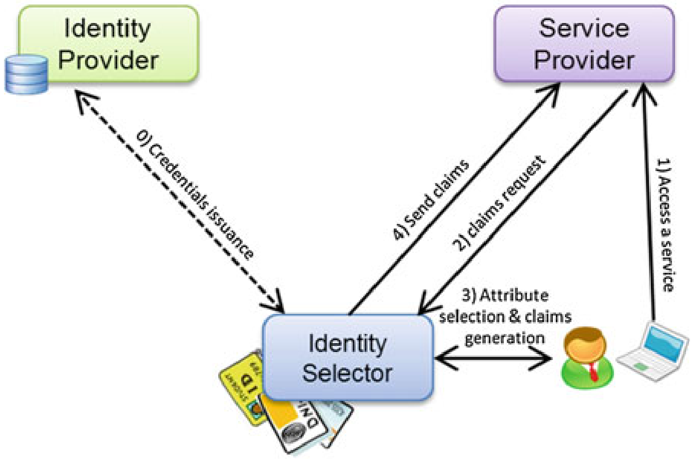


Fig. 10 Idemix general workflow

users in such a way that the identity provider is not required when presenting or validating some users' attributes.

In contrast to U-Prove, Idemix fulfills the selective disclosure of attributes requirement. As shown in Fig. 10, Idemix not only allows the end users to select which attributes will be released to a service provider, but it also has the ability to prove inequalities, such as the value of a birth date attribute being less than today's date minus 21 years without disclosing that birth date attribute value. Idemix could also prove that two attributes have the same value without disclosing that value. However, it neither allows attribute aggregation from different identity providers.

In addition, Idemix tokens are self-blindable. In this way, the end user could transform the token so that they look different each time they need to present and prove some attribute. Therefore, the service provider could not trace end user's accesses. However, this makes the issue of attribute revocation even harder to achieve, which in this case is not available neither for end users, nor for identity providers.

The specification of Idemix presents the usage of short-term claims to replace the revocation behavior. That is, using claims with short expiration time so they need to be renovated every so often. However, although this alternative is valid for certain scenarios, it presents some drawbacks regarding a real revocation mechanism, and some research is currently underway to solve this issue.

5.8 Discussions

This section summarizes the features and limitations of the previously described solutions regarding the requirements presented in Sect. 4.1. Although all solutions fulfill some essential requirements, such as Single Sign-On, confidentiality and integrity requirements, one of the main conclusions is that there is not an ideal approach fulfilling all the requirements. Instead, selecting the most appropriate solution depends on the features of the scenario and the desired behavior.

In general, solutions which do not allow end users to directly control their information are easier to apply, such as SAML, OpenID and OAuth, since the identity providers are in charge of managing such information on their behalf. However, these solutions are based redirections of the end users whenever authentication is required, and this authentication is usually based on username/password. Hence, they are exposed to impersonation if a malicious service provider redirects the end users to a fake identity provider.

On the other hand, although systems which allow end users to control their information usually present user-friendly interfaces, they often require end users to install and manage some applications, and maintain their credentials manually. On the contrary, they use stronger authentication mechanisms, avoiding impersonation.

Additionally, some of the presented systems could be exposed to other security threats, such as the Man-in-the-middle attack, or session related attacks [39] if they do not deploy the appropriate counter-measures. SAML and OpenID standards indicate that the messages must be digitally signed and uniquely identified, to avoid malicious users to modify or replace an assertion, although OAuth just relies on the SSL/TLS channel to exchange messages so no vulnerability establishing such channel could compromise the security of the whole system. CardSpace and Higgins protect the assertions (i.e. tokens indicating authentication statements or attributes) using signatures and a secure communication channel. However, how to acquire Information cards depends on the identity provider, which could raise some security risks if it does not take possible vulnerabilities into consideration.

Regarding privacy, SAML and OAuth make use of pseudonyms to hide the real identity of the end users, but they do not support minimal disclosure information. In other words, the end users could hardly decide which attributes will be exchanged. Yet, OpenID, CardSpace, Higgins and U-Prove do allow the end users to select which attributes will be released, though service providers could trace end user accesses since they use the same pseudonym to access different services or even to access the same service several times. Idemix has the ability of selecting the attribute to be disclosed [3], while presenting each service provider a different identifier, being really hard for them to trace end users accesses. Furthermore, Idemix could make claims based on attributes, so the attributes are not revealed but information based on them instead.

In order to preserve privacy, some systems avoid direct communication between service providers and identity providers, so the latter could not trace end users' accesses. However, that could result in a tough implementation of other features or requirements. For example, OAuth and Higgins issue authorization tokens to allow the service providers to directly access the user information under certain conditions, instead of sending the information directly into the token. Hence, the identity provider and the service provider could exchange information even if the end users go offline.

Furthermore, in solutions like U-Prove or Idemix, where end users can present attributes without involving the identity provider, attribute revocation is hard to achieve. Additionally, since the identity provider cannot trace end users accesses, and the end users are completely anonymous to the service provider, it is difficult to provide these systems with accurate audit mechanisms.

Nevertheless, some of the defined requirements are not properly managed by any of the presented systems, like attribute aggregation. Thus for instance, SAML, OpenID and OAuth are focused on having a unique identity provider for managing all identity-related information of the end users, so attribute aggregation is not contemplated. In turn, CardSpace, Higgins, U-Prove and Idemix support credentials and attributes from different identity providers, for instance, having an Information card from each of them. However, they do not allow presenting information asserted by different providers at the same time.

It is also worth mentioning other trust aspects. Identity management systems assume that trust relationships are established, so they usually require the end users attributes to be asserted by a reliable entity (e.g. a trustworthy identity provider). OpenID, CardSpace and Higgins allow end users to assert self-attributes without requiring them to be validated by a trusted party, which could be useful for non-critical scenarios. However, all the presented identity management systems need additional considerations when deployed on more dynamic environments. Additionally, although end users could approve transactions before releasing any private data in some of the systems, they are not informed about the reliability of the service provider. That is to say, whether the service provider is trustworthy enough to obtain their sensitive information or to provide the expected service.

Table 2 presents a comparative of the current identity management solutions, showing how these solutions meet the aforementioned requirements.

6 Visionary Thoughts for Practitioners

This section sketches the envisioned practical and realistic scenarios of application of advanced identity management techniques within the scope of cloud computing.

In the field of cloud computing, new risks are continuously emerging due to the fact that the number of devices acting in the system is growing exponentially. In other words, the more devices deployed in the system, the more harmful some kind of attacks could be. Furthermore, the number of malwares is also dangerously increasing, since more malicious users could join the system, and new kind of attacks arise.

Although we are talking about a "new" technology or concept, most of the challenges related to cloud computing are not actually new. That is, cloud computing is not something really new, but it rather consists of an integration of technologies related to other contexts, such as multi party computation [5], distributed systems [54], etc. Therefore, some challenges, such as privacy, secure data management, network accessibility, etc. can be handled in the same way as they have been managed in other contexts.

Table 2 Comparative of current identity management solutions within the context of cloud computing

Req.	SAML	OpenID	OAuth	CardSpace	Higgins	UProve	Idemix
R1	Successfully achieved	Successfully achieved	Successfully achieved	Successfully achieved	Successfully achieved	Successfully achieved	Successfully achieved
R2	Successfully achieved	Successfully achieved	Successfully achieved	Successfully achieved	Successfully achieved	Successfully achieved	Successfully achieved
R3	IdPs could log end users accesses	OpenID providers could log end users accesses	OAuth server could log end users accesses	IdPs could log end users accesses	IdPs could log end users accesses	Hard to achieve efficient auditing due to the offered unlinkability properties	Hard to achieve efficient auditing due to the offered unlinkability properties
R4	Authentication mechanism depends on the IdP, although username/pass word is usually used	Authentication mechanism depends on the IdP, although username/pass word is usually used	Authentication mechanism depends on the OAuth server, although username/pass word is usually used	Information cards provide a strong authentication mechanism	Information cards provide a strong authentication mechanism	Authentication based on cryptographic techniques	Authentication based on cryptographic techniques
R5	Static trust relationships are supposed between IdP and SP	RPs do not know the reliability of the IdPs. IdPs and users do not know the reliability of the RPs either	RPs do not know the reliability of the IdPs. IdPs and users do not know the reliability of the RPs either	IdPs cannot prevent the user from sending cards to untrustworthy sites. Users cannot know how reputable a site is	IdPs cannot prevent the user from sending cards to untrustworthy sites. Users cannot know how reputable a site is	IdPs cannot prevent the user from sending cards to untrustworthy sites. Users cannot know how reputable a site is	IdPs cannot prevent the user from sending cards to untrustworthy sites. Users cannot know how reputable a site is
R6	Do not ask for user consent	Explicitily ask for user consent before releasing any user's attribute	Explicit user consent before releasing any user's attribute	The user selects the information card to be sent.	The user selects the card to be sent.	The user selects which attributes will be released	The user selects which attributes will be released
R7	Directly managed by the IdP, not by end users	Directly managed by the IdP, not by end users	Directly managed by the IdP, not by end users	End users store and manage their Information cards by themselves	End users store and manage their Information cards by themselves	End users store and manage their attributes by themselves	End users store and manage their attributes by themselves
R8	Do not require technical knowledge, information is managed by the IdP	Do not require technical knowledge, information is managed by the IdP	Not required technical knowledge. Friendly web pages are usually shown	Requires end users to manually manage their different identities. They even need to make backup of the cards	End user needs to install applications. Similar to Cardspace, it is not trivial for inexpert users	Requires end users to manually manage their attributes. It is not trivial for inexpert users	Requires end users to manually manage their attributes. It is not trivial for inexpert users
R9	Offline mode is not defined in the standard	User interaction is needed tocomplete the information exchange process	Information exchange is directly done between the parties once the authorization	End users interaction is needed toshare their information	Using R-cards the information exchange could be done without end users interaction	End users interaction is needed toshare their information	End users interaction is needed toshare their information

(continued)

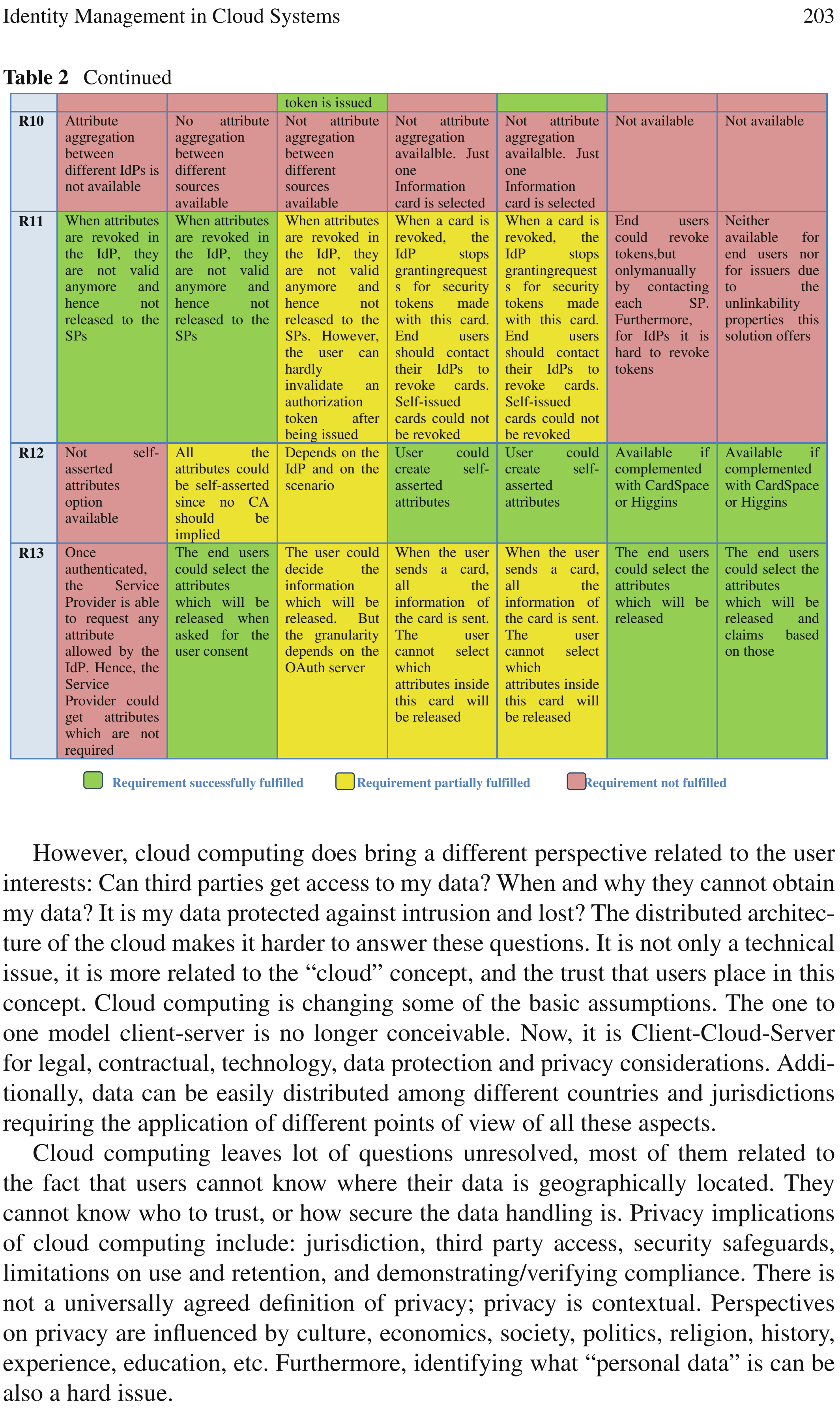


Table 2 Continued

	🟥	🟥	🟩 token is issued	🟥	🟩	🟥	🟥
R10	🟥 Attribute aggregation between different IdPs is not available	🟥 No attribute aggregation between different sources available	🟥 Not attribute aggregation between different sources available	🟥 Not attribute aggregation availalble. Just one Information card is selected	🟥 Not attribute aggregation availalble. Just one Information card is selected	🟥 Not available	🟥 Not available
R11	🟩 When attributes are revoked in the IdP, they are not valid anymore and hence not released to the SPs	🟩 When attributes are revoked in the IdP, they are not valid anymore and hence not released to the SPs	🟩 When attributes are revoked in the IdP, they are not valid anymore and hence not released to the SPs. However, the user can hardly invalidate an authorization token after being issued	🟨 When a card is revoked, the IdP stops grantingrequests for security tokens made with this card. End users should contact their IdPs to revoke cards. Self-issued cards could not be revoked	🟨 When a card is revoked, the IdP stops grantingrequests for security tokens made with this card. End users should contact their IdPs to revoke cards. Self-issued cards could not be revoked	🟥 End users could revoke tokens,but onlymanually by contacting each SP. Furthermore, for IdPs it is hard to revoke tokens	🟥 Neither available for end users nor for issuers due to the unlinkability properties this solution offers
R12	🟥 Not self-asserted attributes option available	🟩 All the attributes could be self-asserted since no CA should be implied	🟨 Depends on the IdP and on the scenario	🟨 User could create self-asserted attributes	🟨 User could create self-asserted attributes	🟩 Available if complemented with CardSpace or Higgins	🟩 Available if complemented with CardSpace or Higgins
R13	🟥 Once authenticated, the Service Provider is able to request any attribute allowed by the IdP. Hence, the Service Provider could get attributes which are not required	🟩 The end users could select the attributes which will be released when asked for the user consent	🟨 The user could decide the information which will be released. But the granularity depends on the OAuth server	🟨 When the user sends a card, all the information of the card is sent. The user cannot select which attributes inside this card will be released	🟨 When the user sends a card, all the information of the card is sent. The user cannot select which attributes inside this card will be released	🟩 The end users could select the attributes which will be released	🟩 The end users could select the attributes which will be released and claims based on those

🟩 Requirement successfully fulfilled 🟨 Requirement partially fulfilled 🟥 Requirement not fulfilled

However, cloud computing does bring a different perspective related to the user interests: Can third parties get access to my data? When and why they cannot obtain my data? It is my data protected against intrusion and lost? The distributed architecture of the cloud makes it harder to answer these questions. It is not only a technical issue, it is more related to the "cloud" concept, and the trust that users place in this concept. Cloud computing is changing some of the basic assumptions. The one to one model client-server is no longer conceivable. Now, it is Client-Cloud-Server for legal, contractual, technology, data protection and privacy considerations. Additionally, data can be easily distributed among different countries and jurisdictions requiring the application of different points of view of all these aspects.

Cloud computing leaves lot of questions unresolved, most of them related to the fact that users cannot know where their data is geographically located. They cannot know who to trust, or how secure the data handling is. Privacy implications of cloud computing include: jurisdiction, third party access, security safeguards, limitations on use and retention, and demonstrating/verifying compliance. There is not a universally agreed definition of privacy; privacy is contextual. Perspectives on privacy are influenced by culture, economics, society, politics, religion, history, experience, education, etc. Furthermore, identifying what "personal data" is can be also a hard issue.

There are other issues focused on enterprises. The enterprises' internal data now could be moved outside the bounds of the company, increasing the need for secure collaborations. Additionally cloud-based environments make policies harder to manage, since it is difficult to answer who has access to what. An issue we should not neglect as well is how organizations using cloud computing can ensure compliance with the laws and regulations that they are subject to.

Cloud service providers should be able to assure that tenants' compliance and security policies are consistently managed and enforced. The identities may need to be governed or managed by geographical locations to enforce regional compliance policies. In the same way, every action that affects a resource being governed by a compliance policy must be recorded. The consumers of the cloud are responsible for the security and integrity of their own data, even when it is held by a (cloud) service provider.

In addition to that, industry and government do not always perceive risks in the same way; therefore, solutions are not equally taken into consideration. On the one hand, industry can see risks as business risks, that is, taking security measures into consideration costs money, so the equilibrium point between not taking security measures at all (zero costs, but maximum risk) and covering all the security issues (maximum cost, minimum risks) should be found. On the other hand, government must avoid any kind of risk, since risks could derive in threats for identity, which may affect the national security. Sometimes solving these issues is not a technical question, but rather a legal one. Additionally, governments may establish clear responsibility lines, which is not a trivial issue in cloud environment. Both industry and government should promote security as an integral part of the technology, not as an extra cost.

Standards can help to address these baseline issues, and could establish the basis for systematic assessment of identity management requirements for cloud systems. They separate technical aspects from legal aspects, describing taxonomy of categories that allows an easier understanding by vendors and customers alike, entailing good practice around category requirements. Common categories emerge from regulations across geographies. Within these categories, regional and national governments have their own identity requirements. However, from regulators' point of view, there are so many standards development, so they do not know where to focus their efforts. They also see a lack of appropriate expertise.

Creating standards or specifications is not an easy task. Currently, we see a lot of concurrent specifications with the same objective, for instance securing assessing, certifying and accrediting cloud solutions. As a result, it is difficult not only to know what we should use, but also some of these specifications contradict each other. Consequently, we cannot follow all of them simultaneously, and it could be a nightmare to try to map these specifications.

Another point to take into account regarding standardization is the implementation of these standards. Usually, standards define how the outcome looks like, but no how to reach it. Therefore, sometimes they are difficult to develop. One of the objectives is also to provide a standards-based framework that will help business process engineers, IT analysts, architects, and developers to implement privacy and security policies in their operations. The PMRM (Privacy Management Reference Model)

[42] provides a guideline for developing operational solutions to privacy issues. It also serves as an analytical tool for assessing the completeness of proposed solutions and as the basis for establishing categories and groupings of privacy management controls.

Cloud computing has to consider current technologies too in order to benefit from them. PKI infrastructures have been used in several solutions until now, and it is a model which has been well accepted for authentication and adapted to different requirements. In the same way, this model is desirable in cloud computing. Organizations, governments and citizens will be eager to use their identities for authentication [58]. Actually, many states are deploying national electronic IDs, so the objective now is looking for ways to leverage existing ID infrastructures.

Authentication (you are who you say you are) is not enough to achieve emerging objectives, so it has been complemented with authorization (you are allowed or not to perform a certain action). XACML [43] is a standard for defining access control policies, which allows fine granularity. Furthermore, it defines a protocol to query for authorization decisions, which allows externalizing authorization management.

The distributed nature of cloud computing permits multiple Identity Providers authentication and authorization services, each of one using different identity credentials, representation and formats, and all of this should be integrated in the same "cloud" concept.

7 Future Research Directions

This section will extract the current research challenges to be addressed in order to reach the aforementioned foreseen practical scenarios. Many of the challenges that cloud computing brings are already handled, such as virtualization [14], federated identity [39], remote data storage accesses [31], etc. But others have not been considered so much yet.

There are still so many unresolved questions regarding data ownership and its access. Cloud computing relieves management of distributed data among different organizations, domains or even countries. In this sense, users' privacy, personal data management and users' rights become more important and they should be carefully considered.

Due to the user-centricity and privacy-preserving features offered by identity management systems, they are becoming a key element in cloud computing environments. Current researches are especially focused on systems based on zero-knowledge proofs, due to the multiple possibilities they may offer. However, as shown in previous sections, some essential features or requirements are still missing in these systems.

Furthermore, on systems based on zero-knowledge proofs, audit and privacy seem to be opposed features. A main goal of identity management systems is to preserve users' privacy by hiding their real identity and interactions, but at the same time they should incorporate effective auditing mechanisms to guarantee that end users cannot deny having performed an operation [59]. Hence, it is a current challenge to define how providers can discourage attackers by recording their actions without tracing end users operations.

There are on-going works taking into consideration this issue [11]. They make use of advanced cryptography techniques extending the anonymous credentials in such a way that the anonymity could be broken under certain conditions. For instance, the identity of a given end user is hidden unless an external and trustworthy inspector (e.g. a government agent) considers that the end user has had a malicious behavior, like committing fraud.

In the same line, since end users could present assertions without requiring involving the identity providers, ordinary Certificates Revocation Lists (CRLs) cannot be used, making efficient attribute revocation hard to achieve. Although some solutions have been already proposed they tend to be impracticable for large systems, and this issue is still an open research question today.

As shown in Sect. 5.8, attribute aggregation is a requirement not properly fulfilled by any of the analyzed solutions. There are some research documents, like [12], proposing solutions for this challenge. Those solutions are usually based on adding an intermediate element in charge of collecting attributes from different providers, but they require end users to manually link their different accounts somehow. Furthermore, how to integrate this kind of solutions within identity management systems based on zero-knowledge proofs, or how they could be adapted to cloud environment has not been still properly faced.

Deployment of cloud computing depends on the features offered to end users to manage their information. End users would be more reluctant to use cloud computing services if they lose the capability of controlling their private information. In this way, user consent is not only a matter of allowing end users to approve whether to continue or not a transaction, but also on defining transitive permission to third parties to access their information. Moreover, identity management in cloud computing is not only authentication any longer, but it also includes authorization. In this sense, existing authorization mechanisms have to be improved or adapted to take into account the dynamicity of cloud computing.

The main difference between traditional systems and cloud computing is that confidentiality based on encryption is hardly possible, and the inexistent user control on the physical level. This affects directly to approaches based on electronic Identity Cards (eID), which now must be “cloud compatible”. For instance, how can cloud fit in projects like STORK [52], which consists of an interoperability framework on top of national eID infrastructure, it is a question that still needs to be answered.

Additionally, identity management systems are usually based on trust relationships between entities or domains. Trust management is an important aspect of identity management, since it defines security boundaries. In this sense, not only service providers could validate end users’ attributes, but also end users (or identity providers acting on their behalf) could determine whether the receiver of their personal information is trustworthy.

However, due to the heterogeneity and dynamicity of cloud computing, trust relationships based on strong contracts, such as SLAs (Service-Level Agreement) establishment, is no longer an option for many scenarios, and more adaptable methods need to be applied. In this regard, reputation systems propose an efficient alternative to handle this issue, although they need to take into account the risk and threats raised by cloud computing. Reputation systems make use of past experiences to calculate a level of trust for a given service, determining whether the service is asserted to be reliable.

8 Conclusion

Identity management systems have been proved to be secure and efficient in diverse contexts and scenarios. By establishing trust relationships between providers and domains, identity management systems offer a huge range of features both for end users and for organizations regarding controlling and exchanging identity-related information in a privacy-aware way.

Due to the user-centricity and privacy-preserving features offered by identity management systems, they are becoming a key element in cloud computing environments. Cloud computing integrates technologies and concepts from other fields, such as multi party computation, distributed systems, federation, etc. Therefore, some of the raised challenges have been already tackled in other contexts, where identity management systems have been widely accepted.

Nevertheless, cloud computing brings a different perspective related to the end users interests, which results on new risks for end users identities. Additionally, end users are more concerned about how their data is managed, where it is located and who can access it. In this sense, cloud computing is changing some of the basic assumptions.

In this document we extract essential requirements as a result of analyzing different use cases and scenarios related to the cloud. As main contribution we have presented representative identity management standards, technologies and approaches in order to highlight the benefits and drawbacks of each of them with regard to the previously presented requirements. An analysis and comparison have been conducted to describe how each of these systems fits in a cloud computing environment.

We have shown that there is not an ideal approach fulfilling all the requirements, which emphasizes all the unsolved questions these systems leave. In this way, selecting the most appropriate solution depends on the features of the scenario and the desired behavior.

In general, this document draws the envisioned practical and realistic scenarios of application of advanced identity management techniques within the scope of cloud computing. Finally, current research challenges to be addressed and ongoing work are presented, together with a description of working groups, standardization activities and international projects aimed to identify, analyze, and describe identity management challenges and pending issues.

References

1. ABC4Trust. Attribute-based credentials for trust. European union funded project of the 7th framework programme. [Online]. Available: https://abc4trust.eu/
2. Alrodhan WA, Mitchell CJ (2007) Addressing privacy issues in CardSpace. In: Proceedings of the 3rd international symposium on information assurance and security (IAS '07), Manchester, UK, pp 285–291
3. Ates M, Buccafurri F, Fayolle J, Lax G (2012) A warning on how to implement anonymous credential protocols into the information card framework. Int J Inf Secur 11(1):33–40
4. Bertocci V, Serack G, Baker C (2008) Understanding windows CardSpace: an introduction to the concepts and challenges of digital identities. Addison-Wesley, Reading
5. Bogdanov D, Niitsoo M, Toft T, Willemson J (2012) High-performance secure multi-party computation for data mining applications. Int J Inf Secur 11(6):403–418
6. Brands S (2000) Rethinking public key infrastructures and digital certificates: building in privacy. MIT Press, Cambridge
7. Brands S, Demuynck L, De Decker B (2007) A practical system for globally revoking the unlinkable pseudonyms of unknown users. In: Proceedings of the 12th Australasian conference on information security and privacy, ACISP'07. Springer
8. Callegati F, Cerroni W, Ramilli M (2009) Man-in-the-middle attack to the HTTPS protocol. IEEE Secur Priv 7(1):78–81
9. Camenisch J, Lysyanskaya A (2001) An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In: Birgit Pfitzmann (ed) Proceedings of the international conference on the theory and application of cryptographic techniques: advances in cryptology (EUROCRYPT '01), Springer-Verlag, London, UK, pp 93–118
10. Camenisch J, Van Herreweghen E (2002) Design and implementation of the idemix anonymous credential system. In: Proceedings of the 9th ACM conference on computer and communications, security
11. Camenisch J, Krontiris I, Lehmann A, Neven G, Paquin C, Rannenberg K, Zwingelberg H (2011) D2.1 architecture for attribute-based credential technologies. Deliverable of ABC4Trust European project
12. Chadwick DW, Inman G (2009) Attribute aggregation in federated identity management. IEEE Comput Soc 42(5):33–40
13. Chappell D (2006) Introducing windows CardSpace. MSDN, Available: http://msdn.microsoft.com/en-us/library/aa480189.aspx
14. Christodorescu M, Sailer R, Schales DL, Sgandurra D, Zamboni D (2009) Cloud security is not (just) virtualization security: a short paper. In: Proceedings of the 2009 ACM workshop on cloud computing security (CCSW '09), ACM, New York, NY, USA, pp 97–102
15. Clercq JD (2002) Single sign-on architectures. In InfraSec '02: proceedings of the international conference on infrastructure security, Springer, Bristol, UK, pp 40–58
16. van Delft B, Oostdijk M (2010) A security analysis of OpenID. Policies Res Identity Manag 343:73–84
17. Dólera Tormo G, Gómez Mármol F, Martínez Pérez G (2012) On the application of trust and reputation management and user-centric techniques for identity management systems. XII Spanish meeting on cryptology and information security (RECSI 2012), San Sebastián, Spain
18. Dólera Tormo G, López Millán G, Martínez Pérez G (2013) Definition of an advanced identity management infrastructure. Int J Inf Secur 12(3):173–200
19. Eclipse.org, Higgins 2.0 Personal Data Service. [Online]. Available: http://www.eclipse.org/higgins/
20. Erdos M, Cantor S (2002) Shibboleth architecture DRAFT v05. [Online]. Available: http://shibboleth.internet2.edu/docs/draft-internet2-shibboleth-arch-v05.pdf
21. Gajek S, Schwenk J, Steiner M, Xuan C (2009) Risks of the CardSpace protocol. Lect Notes Comput Sci 5735:278–293
22. Goldreich O, Micali S, Wigderson A (1991) Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. J ACM (JACM) 38(3):690–728

23. Goldwasser S, Micali S, Rackoff C (1989) The knowledge complexity of interactive proof systems. SIAM J Comput 18(1):186–208
24. Mármol Gómez F, Martínez Pérez G (2009) Security threats scenarios in trust and reputation models for distributed systems. Comput Secur 28(7):545–556
25. Mármol Gómez F, Girao J, Martínez Pérez G (2010) TRIMS, a privacy-aware trust and reputation model for identity management systems. Comput Netw 54(16):2899–2912
26. Gómez Mármol F, Kuhnen M, Martínez Pérez G (2011) Enhancing OpenID through a reputation framework. In: Proceedings of the 8th international conference on autonomic and trusted, computing ATC11, p 118
27. Hammer-Lahav, E. and Recordon, D., "The OAuth 1.0 Protocol", Internet Engineering Task Force (IETF) RFC 5849, 2010.
28. Harding P, Madsen P, Drake TC, Mortimore C (2012) System for cross-domain identity management: core schema. Internet Draft. draft-ietf-scim-core-schema-00 (SCIM)
29. Hardt D (ed) (2012) The OAuth 2.0 authorization framework. Technical report, IETF. Available: http://tools.ietf.org/html/draft-ietf-oauth-v2-31
30. Herranz J, Iñigo J, Pujol H (2009) Privacy features of authentication systems. In: Proceeding of the first workshop on law and web 2.0, Barcelona, Spain. pp 35–46
31. Hoschek W, Jaen-Martinez J, Samar A, Stockinger H, Stockinger K (2000) Data management in an international data grid project. Lect Notes Comput Sci 1971:77–90
32. IBM Research, Zurich (2010) Specification of the identity mixer cryptographic library
33. Identity Commons. [Online]. Available: http://www.identitycommons.net/
34. Jagatic TN, Johnson NA, Jakobsson M, Menczer F (2007) Social phishing. Commun ACM 50:94–100
35. OASIS Standard (2009) Identity Metasystem Interoperability Version 1.0 (IMI 1.0). Available: http://docs.oasis-open.org/imi/identity/v1.0/identity.html
36. Kantara Initiative. [Online]. Available: http://kantarainitiative.org/
37. Kolšek M (2002) Session fixation vulnerability in web-based applications. ACROS security, Available: http://www.acrossecurity.com/papers/session_fixation.pdf
38. Kontaxis G, Polychronakis M, Markatos EP (2012) Minimizing information disclosure to third parties in social login platforms. Int J Inf Secur 11(5):321–332
39. Maler E, Reed D (2008) The venn of identity: options and issues in federated identity management. IEEE Secur Priv 6:16–23
40. Nanda A, Jones MB (2008) Identity selector interoperability profile v1.5. Microsoft Corp. Available: http://download.microsoft.com/download/1/1/a/11ac6505-e4c0-4e05-987c-6f1d31855cd2/Identity_Selector_Interoperability_Profile_V1.5.pdf
41. OASIS IDCloud TC. OASIS identity in the cloud TC. [Online]. Available: http://wiki.oasis-open.org/id-cloud/
42. OASIS Privacy Management Reference Model (PMRM) TC [Online]. Available: http://www.oasis-open.org/committees/pmrm
43. OASIS Standard. eXtensible access control markup language TC v2.0 (XACML) (2005) Available: http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf
44. OASIS Standard: assertions and protocols for the OASIS security assertion markup language (SAML) version 2.0 (2005).
45. OAuth Community. [Online]. Available: http://oauth.net/community/
46. OpenID Community. [Online]. Available: http://openid.net/community/
47. Paquin C, Thompson G (2010) U-prove CTP white paper. Microsoft Tech Rep
48. Pearson S, Benameur A (2010) Privacy, security and trust issues arising from cloud computing. In: Proceedings of the second international conference on cloud computing technology and science (CloudCom), Bristol, UK, pp 693–702
49. PrimeLife. European union funded project of the 7th framework programme. [Online]. Available: http://primelife.ercim.eu/
50. Recordon D, Drummond R (2006) OpenID 2.0: a platform for user-centric identity management. In: Proceedings of the second ACM workshop on digital identity management, Alexandria, VA, USA, pp 11–16

51. Saldhana A, Nadalin A, Rutkowski M (2012) Identity in the cloud use cases version 1.0. Available: http://docs.oasis-open.org/id-cloud/IDCloud-usecases/v1.0/cn01/IDCloud-usecases-v1.0-cn01.html
52. STORK (Secure idenTity acrOss boRders linKed), European Union funded project of the 7th framework programme. [Online]. Available: https://www.eid-stork.eu/
53. SWIFT. Secure widespread identities for federated telecommunications. European Union funded project of the 7th framework programme. [Online]. Available: http://www.ist-swift.org/
54. Tanenbaum AS, Van Steen M (2001) Distributed systems: principles and paradigms. Prentice Hall, Upper Saddle River, NJ
55. The White House. National strategy for trusted identities in cyberspace (NSTIC). [Online]. Available: http://www.nist.gov/
56. Trevithick P. Relationship cards. Higgins report, 19 Sept 2009. Available: http://www.eclipse.org/higgins/documents/relationship-cards.html
57. U-Prove: Microsoft Corporation Technology (2010) [Online]. Available: http://www.microsoft.com/u-prove
58. Ustaoğlu B (2011) Integrating identity-based and certificate-based authenticated key exchange protocols. Int J Inf Secur 10(4):201–212
59. Wang C, Wang Q, Ren K, Lou W (2010) Privacy-preserving public auditing for data storage security in cloud computing. In: Proceedings of the 29th conference on information communications (INFOCOM'10). IEEE Press, Piscataway, pp 525–533
60. Web Identity Working group. [Online]. Available: http://www.w3.org/2011/08/webidentity-charter.html

2

On the Application of Trust and Reputation Management and User-centric Techniques for Identity Management Systems

Title:	On the Application of Trust and Reputation Management and User-centric Techniques for Identity Management Systems
Authors:	Ginés Dólera Tormo, Félix Gómez Mármol, Gregorio Martínez Pérez
Type:	International Conference
Conference:	XII Spanish Meeting on Cryptology and Information Security (RECSI 2012)
Location:	San Sebastián, Spain
Year:	2012
Month:	September
State:	Published

Table 2: On the Application of Trust and Reputation Management and User-centric Techniques for Identity Management Systems

On the Application of Trust and Reputation Management and User-centric Techniques for Identity Management Systems

Ginés Dólera Tormo
Security Group
NEC Laboratories Europe
Email: gines.dolera@neclab.eu

Félix Gómez Mármol
Security Group
NEC Laboratories Europe
Email: felix.gomez-marmol@neclab.eu

Gregorio Martínez Pérez
Department of Information
and Communications Engineering
University of Murcia
Email: gregorio@um.es

Abstract—**Identity Management systems have been designed to deal with the authentication and authorization process. They enable Single Sign-On, where a user can make use of an unique account to access different services, and preserve users' privacy, maintaining users' attributes on reliable providers. However, current identity management systems still lack in giving control to the users to decide which personal information could be released to a given service. In the same way, they do not inform the users about how their personal information will be dealt once released. In this document we present how trust and reputation management and user-centric techniques can be combined with identity management systems to solve these challenges.**

I. Introduction

In the last years, due to the great success of information system, users exchange information more and more, including private information and personal data. However, these users are rarely aware of how their personal data is being managed, and they do not know who are really allowed to get this information. Additionally, users have to deal with registration procedures each time they want to access a service from a service provider with which they did not interact before. This registration procedures requests information about the users, which in most of the cases is not necessary to the provision of the service itself.

Having information about the users is increasingly considered valuable, even it becomes a target to some organizations for business interests, especially for advertising purposes, or aimed to develop advanced attacks on specific targets based on information collected from them. Some organizations try to collect users' information through registration forms. Users are required to create a new account for each service they want to use, for instance they need to execute a registration process just to write some comments in a blog.

Registration forms usually request users personal information, such as email address or birth date. Moreover, some registration processes collect other private data, which is not actually needed for the provision of the service itself, such as telephone number, hobbies, real name, etc. This not only results in having to remember different usernames and passwords for each service and be subject of receiving spam, but also it threatens users' privacy. For example, users do not usually know if their private information will be used in marketing campaign belonging outside to the service provider they are accessing.

Privacy, and more specifically, having control over the information that other entities can have about oneself, are desired features by users of any communication system. Additionally, these topics are being considered in certain geopolitic environments, such as the European Union, as a right of the users. In this context we find those users who do not want to link their private lives with their interactions in different websites they visit, or those who do not want that information about their preferences and usage profiles to be collected. For instance, reporters who want to denounce situations without being concerned about possible retaliation, soldiers who cannot or should not disclose their geographic location, or simply as a measure of safety for any user of communications over the Internet.

With these assumptions identity management systems (IdM) began to emerge a few years ago, suggesting an alternative to these registration processes. Through the establishment trust relationships between different providers, end users are able to store their attributes in reliable entities, which are in charge of preserving users' privacy.

In this document we present some of the main challenges that current identity management systems should deal with regard to the management capabilities and information that users have about their personal attributes. Our contribution in this document is to describe how trust and reputation management and user-centric techniques can be integrated into these identity management systems in order to solve the described challenges.

II. Current challenges in Identity Management Systems

Identity management systems were designed with the aim of providing an access control architecture, able to preserve users' privacy and enabling Single Sign-On by establishing trust relationships between different organizations.

Shibboleth [1] and Liberty Alliance [2] are widely extended examples of identity management systems. In these systems, users' information is stored on reliable entities, such as their city council or universities, named identity providers. The identity providers are in charge of managing users' identities, releasing just needed information to external entities as shown in Figure 1. The service providers delegate the authentication process to these identity providers, which send required users' information after a successful authentication.

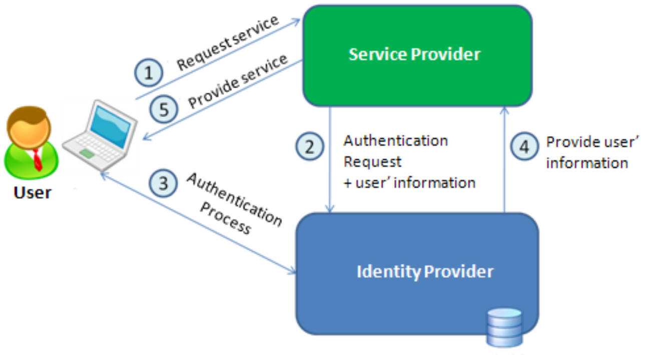


Fig. 1. Overview of an Identity Management System

Since they make use of pseudonyms, identifiable information or attributes, such as email address or real name, does not have to be disclosed if it is not really required. Additionally, it enables Single Sign-On, allowing the users to access different services using their unique account. From the access control point of view, this solution also allows the service providers defining who is allowed to access a specific service through access control rules.

However, these systems do not give enough control to the users for managing their information. Once authenticated they cannot decide which attributes should or should not be released. Similarly, these systems do not give accurate information to the users about how their information will be managed once released.

Current Identity Management systems still present numerous shortcomings, particularly in regard to the control that users have of their own information. In the following we present a set of challenges that this kind of systems needs to address.

- **Let user select data to be released**: The service providers usually require users' attributes either for the provision of the service (e.g. in case they need the post address to deliver something purchased), to give a customized service (e.g. their country to show specific currency), or to perform access control. As previously commented, these attributes are automatically obtained from the identity providers once the user has been authenticated. However, traditional identity management systems rarely allow the user to decide which piece of personal information (or attributes) can or cannot be released to specific entities. Some solutions allow defining which attributes should or should not be released through attribute release policies, although users should manually define complex rules beforehand.
- **Efficient attribute aggregation**: In order for the users to avoid a registration process each time they access a different service provider, they make use of their identity providers to retrieve the required attributes. However, users could belong to different identity providers at the same time. For instance, academic information of a given user could be managed by the identity provider of her university, while information about her post address could be managed by the identity provider of her city hall and their credit card information managed by the identity provider of her bank. In such cases, users have to choose which identity provider to use in order to provide their attributes when requested by a service. This selection depends on the requested attributes, which suppose that users need to have advanced knowledge about how their attributes are spread among the different identity providers. Furthermore, in common identity management systems it is difficult to provide attributes from different identity providers at the same time. For example, if the user needs to provide her credit card number and her post address at once to access a service.
- **Inform the user about the entity she will interact with**: Identity management systems are based on trust relationships between entities. That implies that the identity providers should have established certain agreement with a given service provider before being able to perform any interaction with it. However, the identity providers do not give any information to the users about the service provider. Therefore, the users cannot know how the given service provider behaves before interacting with it, that is, if the service provider will provide the expected service, or even the reliability the user can place in this provider for releasing her attributes. For example, a user accesses a digital library service provider to buy some books, using its identity provider for authentication. The digital library service provider requests the user credit card information which could be provided through the identity provider as well. However the user cannot easily know if the quality of the books is the one expected or even if it is safe for the user to give her credit card information to such a service provider.
- **Hide the accessed services from the identity provider**: Current identity management systems preserve users' privacy, concealing the real identity of the users from the service providers by making use of pseudonyms. When a user accesses a service provider, she presents a signed assertion or a token stating that she has been authenticated by its identity provider, without revealing more personal information. However, since the identity provider has to create these assertions for each service, it can trace users interactions. That is, the identity provider can know which services each of its users has access to.
- **Prevent spoofing and impersonation**: In a common identity management scenario, the service providers redirect the users to their identity providers in order to

delegate the authentication process, also enabling Single Sign-On. This should result in having a secure authentication mechanism, since users should just remember the credentials for a unique account. For instance, if users just have to remember one password it may be a complex and secure one. However, this scenario introduces spoofing since the service provider could redirect the user to a false identity provider, simulating the appearance of the original one, with the aim of collecting users' passwords. Furthermore, if the password of a user is stolen, the malicious entity could both get all the information of the user accessing to her identity provider and impersonate the user accessing other services since it knows the unique password of the user.

- **Trust relationships in dynamic environments**: As previously commented, identity management systems are based on trust relationships between entities. For instance, a service provider accepts authentication assertions from a given identity provider since they trust each other. These trust relationships should be established beforehand based on static agreements, such as SLA (Service Level Agreement). However, in environments where entities are more dynamic, such as in a federation context, and hence the trust relationships are not easy to establish, identity management systems are hard to apply.

Even though these issues have been taken into consideration, they have not been deeply considered in the design process of such solutions. Instead, they have been considered as additional features to improve the behavior of such solutions. Current identity management systems have given higher priority to the fact of having more control over the users, through applying access control policies, deploying mechanisms able to incriminate the users if they perform malicious actions. Nevertheless, current systems have not adequately given control capabilities to the users, with regard to the capabilities of controlling how their information will be managed.

III. User-centric Identity Management techniques

User-centric techniques are referred to those which give extensive attention to the users in the design of a solution. Within the context of identity management, user-centric techniques have been proposed in order to give more control and information to the users about how their information is dealt, while at the same time being compatible with traditional identity management systems.

OpenID [3] is defined as a user-centric identity management system. Its functionality is similar to those presented in the previous section, in the sense that users' attributes are stored in OpenID Providers (i.e. identity providers) and requested by Relying Parties (i.e. service providers) when the user wants to access a service given by a Relying Party. In the context of user capabilities, the main difference between the traditional identity management systems is that the OpenID Providers ask for explicit user consent before releasing any kind of personal information.

In order to prevent spoofing, identity providers within traditional identity management systems could make use of strong authentication mechanisms, such as authentication based on certificates. Using certificates, a user easily realizes if the site of her identity provider has been faked by a malicious entity, since the browser checks the certificate of the site to perform the authentication. Furthermore, even if the user is maliciously authenticated, impersonation could not be possible with this mechanism since the user's private key is not released. Nevertheless, making use of strong authentication mechanisms usually require more comprehensibility from the user point of view, and they are difficult to adapt in environments where technical abilities cannot be supposed from the users.

As an alternative to authentication based on certificates there are the Information cards (I-cards) [4]. Information cards represent personal digital identities, and are maintained by each user. The concept tries to simulate the real identity cards which the users carry in their pockets, such as national id card, driver license, public library member card, etc. These cards may also contains users' attributes, which can be signed by entities (e.g. identity providers) to prove the validity of them. In this sense, the users could choose any card to present when they are accessing a service. Furthermore, the users do not have to access their identity provider to authenticate and get the attributes each time they need to access a service.

Since the Information-cards are maintained by the users, additional tools, such as the Identity Selector [7], have been defined to assist the users in such management. The Identity Selector is in charge of storing, managing and presenting the information to the user of the Information-cards. This Identity Selector could be an application in the user device, able to manage the different Information-cards of the user locally, or it could be used as an external service where the user access, in a secure way, to get one of their identity card when required. Figure 2 shows an overview of an identity management scenario where the user makes use of an Identity Selector to present required attributes.

The Identity Selector also assists the users in the process of selection of cards. According to the attributes that are being requested, the Identity Selector is able to recommend the card to make use of. However, presenting an Information-card could release more attributes than required. For example, if the service provider requires the user post address, the user could present her national id card, but it also could contain other non-required information, such as real name or national identification number. Some improvements to these Information-cards have been proposed to avoid this privacy issue, such as U-prove [6], where the users could generate claims containing just the set of attributes which have been requested.

In case a user wants to present attributes belonging to different entities at the same time, that is, stored in different Information-card, the attributes of this identity cards should be aggregated as previously commented. Some research has been done in this direction, proposing solutions where an identity card is generated from the attributes of other identity cards.

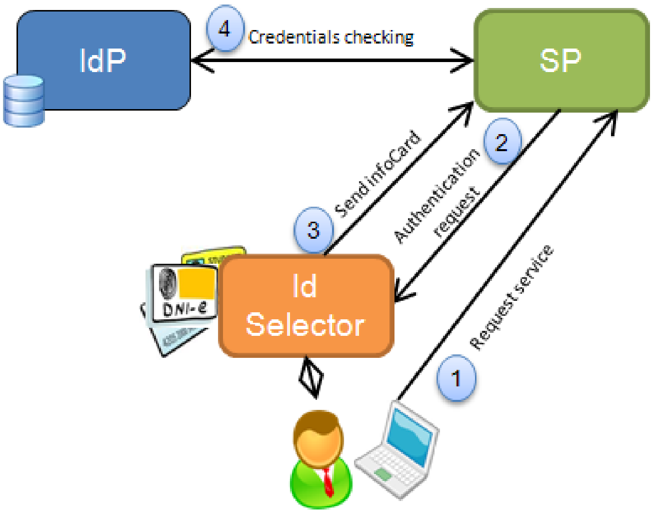


Fig. 2. Overview of a Identity Selector in a identity management system

The Higgins project[5] have designed and implemented some Identity Selectors going in this direction.

In general, even though these system allow the users control which information each entity could have, they do not show if the users can trust in the entity requesting such information, nor how it will be managed. Furthermore, the users cannot know if the requested service will fulfill the expectations of the user.

IV. Trust and Reputation Management

For some years now, trust and reputation management has emerged as a very promising and appealing trend to deal and cope with a number of security threats risking the wide use and deployment of the so-called information and communication technologies.

Thus for instance, very popular sites such as Amazon or eBay have been using this powerful tool since their early conception in order to provider their customers with very valuable information regarding the expected behavior (i.e., reputation) of the participants in their respective systems (sellers, buyers, service providers, etc.).

Indeed, trust and reputation management constitutes a very helpful instrument to identify malicious or selfish elements interacting in certain systems. As a research topic, it has captured the attention of both industry and Academia, leading to a torrent of outstanding results materialized in the form of final products, patents, standards and research articles, amongst others.

Due to its numerous benefits, it has been effectively applied in a multitude of scenarios or environments, raging from P2P networks, wireless sensor networks, vehicular ad-hoc networks, intrusion detection networks, social networks, internet of things, cloud computing, e-Commerce, etc.

Yet, trust and reputation management finds one of its best coupling when it is employed in conjunction with identity management systems. It is in this case where it helps IdM systems to really thrive and move to a next step, fostering in addition their wide social acceptance.

Linking with the challenges presented in section II, trust and reputation management probably represents the most suitable tool to tackle both the disinformation of the users regarding the entities they interact with, as well as the smart establishment of dynamic trust relationships.

To handle the first aforementioned challenge, trust and reputation management systems take care of gathering behavioral information about the target entity (or entities). In most cases such collected information is expressed as recommendations or feedbacks from those users who previously interacted with the target entity. Next, trust and reputation management systems perform an aggregation of such information (or an update of previous data) aiming to obtain an accurate and representative trust and/or reputation score for the target entity. Finally, such information is given to the user, who will be empowered now to make a smarter and safer decision on whether to interact or not with the target entity.

A good example of this advantageous integration was presented in a previous work [8], where an enhancement of the OpenID protocol by means of an accurate and robust trust and reputation management system was described.

On the other hand, the problem of those rigid systems where the establishment of new trust relationships might become a lengthy, hard and even costly issue can be very nicely addressed as well by an efficient trust and reputation management mechanism. This is the case, to name one, of the identity federation scenarios, where several entities collaborate in order to share the users' identity information they handle, for the sake of the whole community.

Nevertheless, trust relationships in these environments have been traditionally based on rigid and most of the times inflexible agreements like SLAs, hindering this way the rapid and dynamic creation, evolution and termination of identity federation systems.

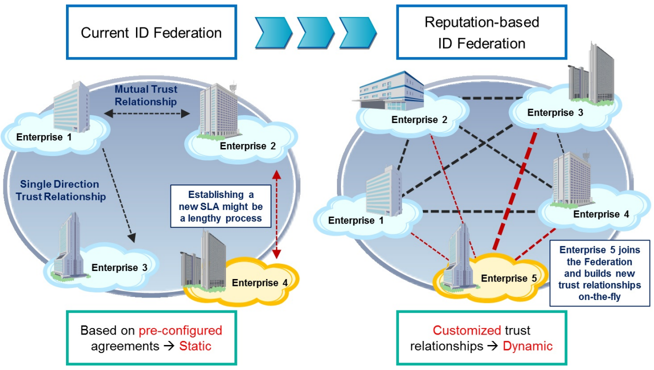

Current ID Federation
Reputation-based ID Federation
Mutual Trust Relationship
Enterprise 1
Enterprise 2
Single Direction Trust Relationship
Establishing a new SLA might be a lengthy process
Enterprise 3
Enterprise 4
Enterprise 2
Enterprise 3
Enterprise 1
Enterprise 4
Enterprise 5
Enterprise 5 joins the Federation and builds new trust relationships on-the-fly
Based on pre-configured agreements → Static
Customized trust relationships → Dynamic

Fig. 3. Reputation-based identity federation

Once again, trust and reputation management brings an elegant solution to this matter, as shown in Figure 3. Here, a new entity willing to enter the federation or become one of its members has the chance to do it in a seamless and dynamic fashion, without the need to trigger a lengthy negotiation process oriented to the acquisition of a SLA. To this end, each of the current members of the federation will assess the trustworthiness of the newcomer and establish new trust relationships on-the-fly, accordingly.

Moreover, those dynamically established links might evolve throughout time based on the behavior (and therefore the associated reputation score) of the new entity, meaning that the other members will exchange more or less information (users' identity attributes) with such entity according to its goodness.

V. Challenges Analysis

In section III we have described how user-centric techniques could resolve some of the challenges presented by current identity management systems regarding to the control given to the users about their private information. These techniques tend to give more selection capabilities to the users in such a way that they can choose which digital identity they want to present to a specific service. One of the main aim of these techniques is also preserve users' privacy, since they are able to release just the needed private information to access a service.

Similarly, section IV describes how trust and reputation management systems could resolve some of the challenges regarding the information given to the users about the services they are accessing to. Before the users release private information to a given service provider, these systems collects recommendations from other users or entities about such a service provider. These recommendations are based on past experiences and they could predict, to some extend, the behavior of the service provider. According to the recommendations, the users can have an idea of the expected service and if they can trust in the service provider. Finally, they could decide if they want to continue (or not) the communication with the service provider, and hence releasing the requested attributes.

The integration of both user-centric techniques and trust and reputation management, within identity management context, could result in improved identity management systems able to give more control and information to the users. Table I summarizes how the combination of both topics would achieve the described challenges.

Challenge	User-centric Techniques	Trust and Reputation Management
Let user select data to be released	Allow selecting user attributes before releasing from the identity provider (e.g. OpenID), select a specific identity (I-Cards) or select a set of attributes (U-Prove)	Non-Applicable
Efficient attribute aggregation	Some solutions allow collecting attributed from different sources before releasing them (Higgins)	Non-Applicable
Inform the user about the entity she will interact with	Not available in current solutions	Collect recommendations about a given service provider, based on past interactions in order to inform the user about the service before accessing it
Hide the accessed services from the identity provider	Some user-centric systems do not require the users to have interactions with the identity provider to access a service provider. Instead, the attributes could be stored in the user device and directly validated by the service provider (e.g. U-Prove)	Non-applicable
Prevent spoofing and impersonation	Information-cards, among others, propose an alternative to passwords preventing spoofing and impersonation. Users do not need to send their passwords through the network nor introduce them in an external website	Trust and reputation systems avoid malicious entities, since they are punished if they are not behaving properly. Users do not accept services of malicious service provider since they get low reputation
Trust relationships in dynamic environments	Require trust relationships previously established	Trust relationships could be established dynamically since they could be based on past interactions

TABLE I
ANALYSIS OF CHALLENGES REGARDING USER-CENTRIC TECHNIQUES AND TRUST AND REPUTATION MANAGEMENT

VI. Conclusion

Identity management systems have been designed with the aim of enabling Single Sign-On and preserving users' privacy. Nevertheless, current identity management systems still present some challenges to be solved, with regard to the management capabilities that the users have about their personal attribute. In the same way, users are not properly informed about who will have access to their data once released. That is, they do not know if they can trust in a given service provider before interacting with it. In this document we have presented some challenges that current systems have to achieve.

User-centric techniques give more selection capabilities to the users in such a way that they can choose which digital identity they want to present to a specific service. Trust and reputation management systems are useful to identify malicious elements interacting in certain systems, especially in environments where strong trust agreements could not be supposed. Since these systems collects recommendations, based on past interactions, they are able to inform the users how a service will be, to some extent, before they interact with it.

We have described how user-centric techniques and trust and reputation systems could be integrated in identity management systems to achieve some of the presented challenges. Finally, we have analyze how these topics could be combined, in order to give more control and information to the users within identity management systems.

References

[1] Erdos, M., Cantor, S.: "Shibboleth-Architecture DRAFT v05", *I*nternet2/MACE 2002.

[2] Wason, T., Alliance, L., Hodges, J., Kemp, J., Thompson, P.: "Liberty Id-FF Architecture Overview", *L*iberty Alliance, 2003.

[3] Recordon, D. and Reed, D.: "OpenID 2.0: a platform for user-centric identity management", *P*roceedings of the second ACM workshop on Digital identity management (2006)

[4] Jones, M.B.: "The identity metasystem: A user-centric, inclusive web authentication solution", *T*oward a More Secure Web-W3C Workshop on Transparency and Usability of Web Authentication (2006)

[5] Higgins Personal Data Service: http://www.eclipse.org/higgins, (2007)

[6] Paquin, C. and Thompson, G.: "U-Prove CTP White Paper", *M*icrosoft Corporation (2010)

[7] Nanda, A. and Jones, M.B.: "Identity selector interoperability profile v1. 5", *M*icrosoft Corporation(2008)

[8] F. Gomez Marmol, M. Kuhnen, G. Martnez Perez : "Enhancing OpenID through a Reputation Framework", *P*roceedings of the 8th international conference on Autonomic and Trusted Computing ATC11, pp. 118 (2011)

3

Towards the Integration of Reputation Management in OpenID

Title:	Towards the Integration of Reputation Management in OpenID
Authors:	Ginés Dólera Tormo, Félix Gómez Mármol, Gregorio Martínez Pérez
Type:	Journal
Journal:	Computer Standards & Interfaces, Special Issue on Secure Mobility in Future Communication Systems under Standardization
Publisher:	Elsevier
Volume:	36
Number:	3
Pages:	438-453
Year:	2014
Month:	March
DOI:	`http://dx.doi.org/10.1016/j.csi.2013.08.018`
State:	Published

Table 3: Towards the Integration of Reputation Management in OpenID

Computer Standards & Interfaces 36 (2014) 438–453

Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi

Towards the integration of reputation management in OpenID

Ginés Dólera Tormo [a,*], Félix Gómez Mármol [a], Gregorio Martínez Pérez [b]

[a] NEC Europe Ltd., Kurfürsten-Anlage 36, 69115 Heidelberg, Germany
[b] Department of Information and Communications Engineering, University of Murcia, Murcia 30100, Spain

ARTICLE INFO

Article history:
Received 15 January 2013
Received in revised form 30 August 2013
Accepted 30 August 2013
Available online 9 September 2013

Keywords:
Single Sign-On
Reputation
OpenID
User-centric
Identity management

ABSTRACT

OpenID is an open standard providing a decentralized authentication mechanism to end users. It is based on a unique URL (Uniform Resource Locator) or XRI (Extensible Resource Identifier) as identifier of the user. This fact of using a single identifier confers this approach an interesting added-value when users want to get access to different services in the Internet, since users do not need to create a new account on every website they are visiting. However, OpenID providers are normally used as a point to store certain personal attributes of the end users too, which might be of interest for any service provider willing to make profit from collecting that personal information. The definition of a reputation management solution integrated as part of the OpenID protocol can help users to determine whether a given service provider is more or less reliable before interacting with it and transferring their private information. This paper is providing the definition of a reputation framework that can be applied to the OpenID SSO (Single Sign-On) standard solution. It also defines how the protocol itself can be enhanced so OpenID providers can collect (and provide) recommendations from (to) users regarding different service providers and thus enhancing the users' experience when using OpenID. Besides the definition, a set of tests has been performed validating the feasibility of the framework.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

Providing effective authentication solutions is a key part to successfully deploy any service provider nowadays. That implies, as a minimum, to identify individuals in such providers and to control the access to the different resources and services being provided by them.

Even if these authentication approaches can be based on well-known technologies such as login/password, smart cards, digital certificates or biometric information, among others, it is usually the case that different service providers belonging to different companies or organizations are managing their own identifiers and mechanisms to authenticate their users [1–4]. This is leading to users creating new accounts on almost every website where they are required to do so and, in certain cases, even avoiding websites where they have to create yet another identifier (e.g., username and password) [5].

OpenID [6,7] is an open technology standard that provides a solution to this problem. As such, it is defined as a mechanism allowing the use of a single account to sign in to different service providers. In this proposal, the user only has to enable her current existing account for OpenID access and then provide any OpenID-enabled service her unique OpenID identifier. With this identifier the service provider redirects the user to the OpenID provider where she can be authenticated and then get access (after successful authentication) to the service.

The wide use of this approach as well as the information that certain service providers are requesting from the users are making OpenID providers the right place to store certain private attributes of the end user. Those attributes are also needed when taking certain decisions in the service provider so the access can be provided (or denied) to particular resources. Such access may depend on the role of the user, the domain where she is coming from, her age, etc.

However, as this private information is directly exchanged between the OpenID provider and the service provider via a set of OpenID extensions and the user is not having direct control on this exchange under certain circumstances, there is a clear need to extend the OpenID standard to provide a tighter control over such exchange.

Several approaches can be considered here, for instance deploying solutions based on pre-established agreements to regulate how the information is released [8], deploying white/black lists [9], or allowing the users to manually manage attribute release policies [10] to decide which entities have access to their attributes. However, due to the dynamic and decentralized nature of OpenID, trust and reputation management becomes a promising option [11,12]. It can provide end users with certain key information before starting an OpenID authentication process (and attribute release) with an unknown service. Users can then decide whether they are willing to exchange this personal information with that service or not, based on the interactions that other users had in the past, i.e., based on the reputation that this service provider is having among different users.

* Corresponding author. Tel.: +49 6221 4342 220; fax: +49 6221 4342 155.
E-mail addresses: gines.dolera@neclab.eu (G. Dólera Tormo), felix.gomez-marmol@neclab.eu (F. Gómez Mármol), gregorio@um.es (G. Martínez Pérez).

0920-5489/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.csi.2013.08.018

This paper provides a detailed definition of a reputation framework designed to be integrated with OpenID. Moreover, it is describing how the OpenID protocol can be enhanced so the OpenID provider can collect recommendations from different users on a given service provider based on their interactions with it, although they belong to other OpenID providers. Our work also describes how these recommendations can be aggregated appropriately and provided to the user before she starts interacting with a service.

This paper is a revised and extended version of a previous publication [13]. The paper at hand includes the description of several reputation computation engines, showing different ways of aggregating recommendations from different sources. In addition, we have defined a simulation environment in order to validate the feasibility of the framework. Not only the behavior of the framework, but also the advantages and disadvantages of different reputation computation engines, are analyzed and explained how they fit to different system conditions.

The remainder of the article is organized as follows. Section 2 provides the main references and related works. Section 3 provides a common nomenclature as well as the description of the particular problem being addressed as part of this research work. Then, in Section 4 we describe both the functional and non-functional requirements for developing a reputation framework, while in Section 5 the OpenID protocol enhancement needed to deal with this reputation framework is presented. Section 6 shows the reputation framework itself, which has been designed for enhancing the users' experience using the OpenID technology. Later, Section 7 explains different ways to aggregate the collected recommendations to compute the reputation value, while Section 8 presents the experiments conducted to validate the feasibility of the proposed framework. Finally, in Section 9 the main conclusions and lines of future work are identified.

2. Related work

Counting on any authentication system with Single Sign-On (SSO) capabilities is an essential characteristic for any Internet service nowadays. Modern solutions establish protocols to exchange authentication data in such a way that the authentication could be delegated to an external service. For example, Kerberos [14] defines tickets allowing users to be authenticated in one place and accessing to different services without requiring them to be continuously validating their credentials. Furthermore, recent works enhance the Kerberos protocol by guaranteeing the anonymity of the users while accessing those services [15].

We focus our contribution on the OpenID protocol [6] because of the challenges it raises due to its decentralized nature. Many authentication systems count on trust relationships, usually maintained by a central authority. This central authority maintains the trust information of all the entities involved in the system. However, in distributed systems, further mechanisms are required to manage the trust, which could be based, among other relevant information, on recommendations and past experiences, managed by a reputation framework [16]. This section describes in more detail the work done in this specific area and contextualizes our work within this field.

Trust and reputation management systems for distributed and heterogeneous environments have been studied since a while [17–19]. Moreover, reputation frameworks have been proposed in different contexts, e.g., P2P file sharing [20–23] and reputation enabled service-oriented frameworks [24–28]. Thereafter, the next trends that we could analyze are the application of reputation frameworks for enhancing authentication systems and the proposal of distributed reputation frameworks.

The TRIMS framework [29] applies a trust and reputation model aiming to guarantee an acceptable level of security when deciding if a service in a different domain is reliable for receiving user's personal data. That is to say, it applies reputation techniques for enhancing the privacy of the users when exchanging attributes between services, in a multi-domain scenario. Each domain relies on its own past experiences with the other domain being evaluated. Those experiences are weighted in order to give more or less importance on the final result.

Following the idea of the TRIMS framework, AttributeTrust Framework [30] deals with the trust of relying parties when requesting user's attributes during Internet transactions. They address the problem by aggregating user's attributes in defined Attribute Providers and then perform policy based trust negotiations for evaluating trust in the attributes. They proposed a reputation model for calculating confidence values that result from confidence paths leading from the relying party to the attribute providers. Such reputation model is resilient to the common attacks known in reputation systems.

Authors of [31] introduce a flexible reputation system framework to augment explicit authorization in a web application. They argue that explicit authorization frameworks implemented with access control lists (ACL), capabilities, or roles (RBAC) [32] require such a high overhead to the administrators for manually granting the user's specific privileges, that it cannot scale for Internet type of applications. The framework supports multiple computation models for reputation. However, different from our framework, its focus and design choices target reputation calculations for human subjects. For example, the framework helps to decide which users' identifiers should a service provider support. Moreover, it is a centralized framework and also does not consider opinions of other users because of the complexity that user's opinions bring to the system. It is an objective reputation system based on measurements.

In [33] authors propose a model for determining the level of trust that a service provider could hold on the subject interacting with it. The trust is based on the reputation values that the users have gained for interacting with the offered services. Aimed at federated environment, this solution maintains the reputation of the users in the Identity Providers. Hence, together with the authentication token, it is sent the reputation value of the user accessing a service.

In [34] authors propose a distributed reputation and trust management framework where trust brokers exchange and collect information data about services. By doing so, individual users only need to ask their brokers for accessing reputation information. They claim that due to the distributed nature of the brokers it is impossible to collect the information of all brokers. From the authors' perspective, every user would have an online trust broker, which would collect reputation information for them. The personal brokers are then hierarchically organized for the information distribution. The approach is based on a global database that has information about all servers. Therefore, there is a centralized component in this distributed approach that can also bring a single point of failure and all the other drawbacks common to centralized systems.

Authors in [35] make use of trusted computing to increase the reliability of the OpenID protocol. It replaces the common username/password authentication mechanism for a trusted platform module (TPM) [36]. In this way, in the authentication process, the OpenID provider can compare the reported system state to previously generated reference values, allowing only trustworthy clients to login and claim an identity.

There are some approaches describing reputation frameworks for OpenID-based solutions [37]. However, these solutions are based on centralized concepts, where a trustworthy entity is in charge of managing the reputation information of all the entities.

Analyzing the aforementioned works, we came to the conclusion that, to the best of our knowledge, there is no related work targeting a distributed reputation framework applied on top of the SSO OpenID protocol that can provide reputation information about the relying parties to the users prior to their interaction. We claim that such distributed framework on top of the OpenID protocol can enhance the user experience when dealing with SSO in the Internet. What we achieve here is a unique framework providing SSO and reputation information at the same time.

3. Problem statement

For consistency throughout the remainder of the paper, we present next a basic glossary of the terms used within the OpenID environment:

- User or end-user: The entity that wants to assert a particular identity.
- Identifier or OpenID: The URL or XRI chosen by the end-user to name the end-user's identity (for instance, http://felixgm.myopenid.com).
- Identity provider or OpenID provider (IdP or OP): A service that specializes in registering OpenID URLs or XRIs and providing OpenID authentication (and possibly other identity services).
- Relying party (RP or SP): The site that wants to verify the end-user's identifier; other terms include "service provider" or the now obsolete "consumer".

Once we have defined the meaning of those terms and the role of each player, we present how they interact in a common scenario.

Let's say Alice wants to watch a film online, so she accesses the service provider (relying party) offering such service. However, the film that Alice wants to watch contains explicit violent scenes and she must therefore prove that she is an adult in order to get access to it.

Then, instead of registering and creating yet another account for this particular RP, Alice wants to use her existing OpenID already registered in a certain OP. If it is allowed by Alice, the RP would have access to Alice's identity information stored in the specified OP (like for instance, age, e-mail, and credit card) and, after checking that she is an adult, the RP actually provides the requested film.

Fig. 1 represents the sequence diagram corresponding to the regular operation of the OpenID protocol. The workflow starts when a user tries to access a protected resource of the relying party (RP), for instance a film in the aforementioned example (step 1). The RP shows to the user an authentication page (step 2), where the user is able to make use of her OpenID URL (step 3) to be redirected to her OpenID provider (step 8). Before being redirected, the RP and the OP start an association process, establishing a shared secret between them, which is used to verify subsequent protocol messages (steps 4 to 7).

When the user is authenticated in her OpenID provider (steps 9 to 11) she is redirected back to the RP (step 12). The RP can then validate the authentication response making use of the shared secret previously established (step 13). Along with the authentication response, some requested user attributes are also sent, if the user has given her consent. Finally, the RP provides the resource of the service that the user has been requested (step 14).

Nevertheless, despite its several advantages, based on a market survey [38], 97% of the users today in the Internet would like their OpenID providers to offer a way of assisting them with trustworthy

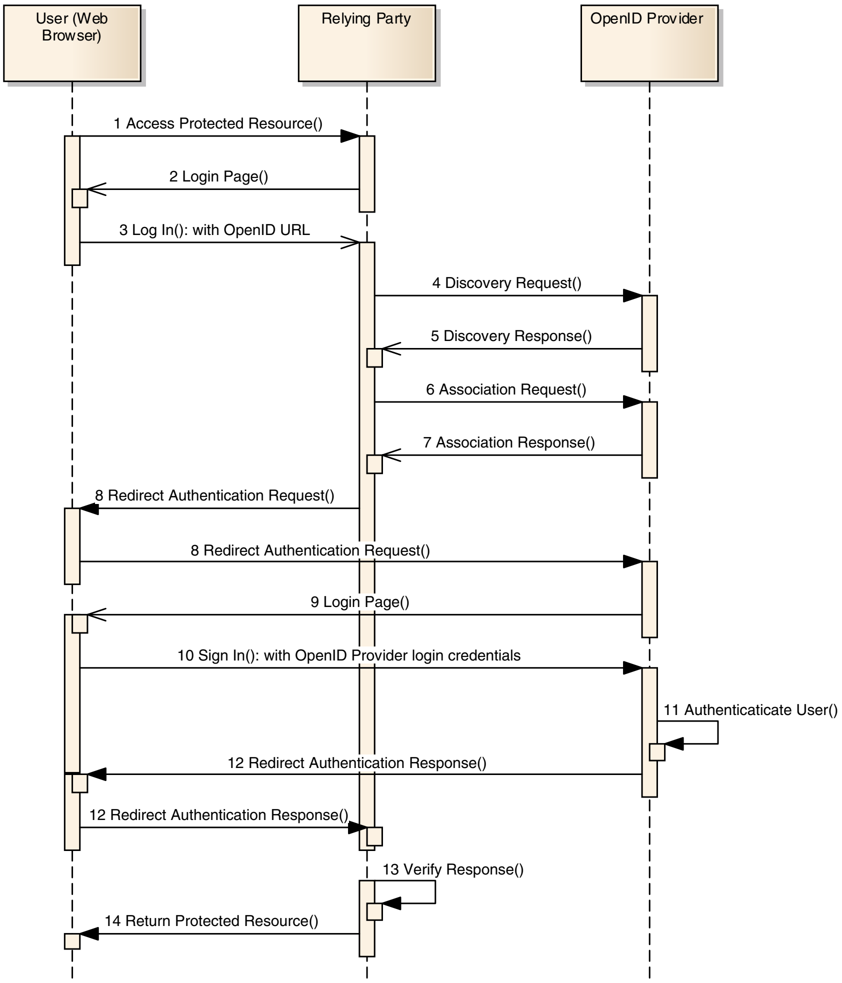

User (Web Browser)
Relying Party
OpenID Provider
1 Access Protected Resource()
2 Login Page()
3 Log In(): with OpenID URL
4 Discovery Request()
5 Discovery Response()
6 Association Request()
7 Association Response()
8 Redirect Authentication Request()
8 Redirect Authentication Request()
9 Login Page()
10 Sign In(): with OpenID Provider login credentials
11 Authenticaticate User()
12 Redirect Authentication Response()
12 Redirect Authentication Response()
13 Verify Response()
14 Return Protected Resource()

Fig. 1. OpenID protocol: sequence diagram.

information about the relying parties (services that they use). Such reputation information would lead to smarter and more accurate decisions from users when deciding which relying parties to interact with, while preventing them from having transactions with malicious or fraudulent RP, which would be, in turn, identified and isolated.

Spiteful relying parties might misbehave and misuse users' personal information like e-mail address for spamming, or credit card number for charging unexpected expenses, amongst many other dishonest operations. Therefore it is crucial to promptly and accurately detect their unreliable behavior and share this information in the form of a low reputation value within the community, in order to warn other (maybe unwary) users.

The main aim and contribution of our work is to enhance such protocol so that the specified OpenID provider is able to collect recommendations about the selected RP, and aggregate them appropriately in order to provide the user with a useful and reliable reputation score about the RP.

4. Requirements analysis

This section identifies the functional as well as non-functional requirements for developing the envisioned reputation framework for the OpenID SSO system. As in any other study of this category, the requirements represent a list of trade-offs that have to be analyzed and evaluated when building such a system. Our proposal is intended to fulfill these requirements, since currently there is not any OpenID implementation which addresses all of them. First, we address the functional requirements that are relevant to the framework. These are:

1. Majority rating evaluation: In order to provide a realistic and fair representation of the RP behavior in terms of a reputation score, our framework should consider the suggestions or feedbacks provided by the majority of raters.
2. Time awareness: Not only the majority has to be considered, but the framework should consider as well that old ratings should be treated as less important than new ones. Therefore the framework should consider the instant when the recommendation ratings were provided to the relying parties.
3. Incorrectness awareness: The framework should consider the possible incorrect feedbacks provided by either malicious users [39] or users that by mistake provide wrong rating values to the relying parties.
4. Users' preferences awareness: The framework should provide a mechanism that allows users to look for services based on their preferences [40,41]. That means, the framework should provide a mechanism where the users can express their preferences with regard to the provision of each service.
5. Privacy: The framework should provide a mechanism allowing users to rate service providers in a privacy-preserving way [42,43]. Only the OpenID provider should know about the digital identity of the user. This should be protected from the relying parties which receive the recommendation information. We believe such mechanism will give an extra incentive to the users for providing feedback information about the relying parties that they have interacted with.

Moreover, we foresee the following non-functional requirements as the most relevant in order to provide a reliable reputation framework on top of a SSO system like OpenID.

6. Scalability: When designing the system, we have to take care of the rate of recommendation inputs and queries made on the system. A centralized or distributed solution might have different implications regarding scalability issues. It is therefore important to bear in mind the potential bottlenecks of the architecture that might also constitute a single point of failure.
7. Reliability of the transaction: We believe that, for certain specific situations (like those with a very high frequency of transactions), a reputation system might not provide a 100% reliable transactional support for users' recommendations input. It should rather consist of a best-effort solution based on messages to be exchanged between the different peers. We foresee a high load of interactions; therefore, a 100% reliable transactional support might give additional delay or even block the system.
8. Performance: We think that the system should support a lot of applications requests at high rates. For example, a popular OpenID provider, which is accessed by a lot of users, will also have to communicate with other OpenID providers in order to exchange reputation information with them. Such exchange of information needs to have a high performance because otherwise the user experience will be degraded.
9. Reputation model: The system should support different reputation models, since we believe those models will be improved, due to a lot of research happening in this area. Therefore, it is important for the framework to be able to support different reputation models on the fly through a reputation model plug-in framework.
10. Portability of data exchange: The framework should allow data describing the reputation information of the relying parties to be exchanged across the different trust management frameworks. At the current state of the art, there is no protocol between OpenID providers allowing the exchange of information between them. Therefore, the framework requires a protocol and a standardized model for reputation data that can be exchanged between OpenID parties [44].
11. Compliance with laws and regulations: Since these SSO protocols might deal with very sensitive and private users' information, any enhancement over them must keep the compliance with current related laws and regulations. Moreover, such compliance with regulations will improve the users' perception of security in the system and, therefore, their willingness to adopt it.

5. OpenID enhancement

As we commented previously, the OpenID protocol defines a decentralized SSO solution, where any entity could form part of the system by only implementing the defined protocol. In this section we present the OpenID protocol enhancement needed to deal with the presented reputation framework.

Based on the common OpenID workflow, Fig. 2 shows a modified sequence diagram. The principal goal of the reputation framework is depicted in steps 11.1 to 11.5. In these steps, the OpenID provider queries other OpenID providers regarding the reputation of the RP (steps 11.1 and 11.2). How the OpenID provider knows which other OpenID providers have interacted with the RP, therefore having updated reputation values, is described in the following sections. Once collected, the OpenID provider calculates an overall reputation (step 11.3), giving the user of such a system the possibility to receive reputation information about the relying party that she is accessing (step 11.4).

As it is currently considered in the OpenID protocol, the user should provide explicit consent to release the requested attributes. With this enhancement, the user is not only informed about the attributes which would be released, but also she could make that decision based on the reputation of the RP. In other words, the user could cancel the operation if she considers that the RP does not have enough reputation (step 11.5).

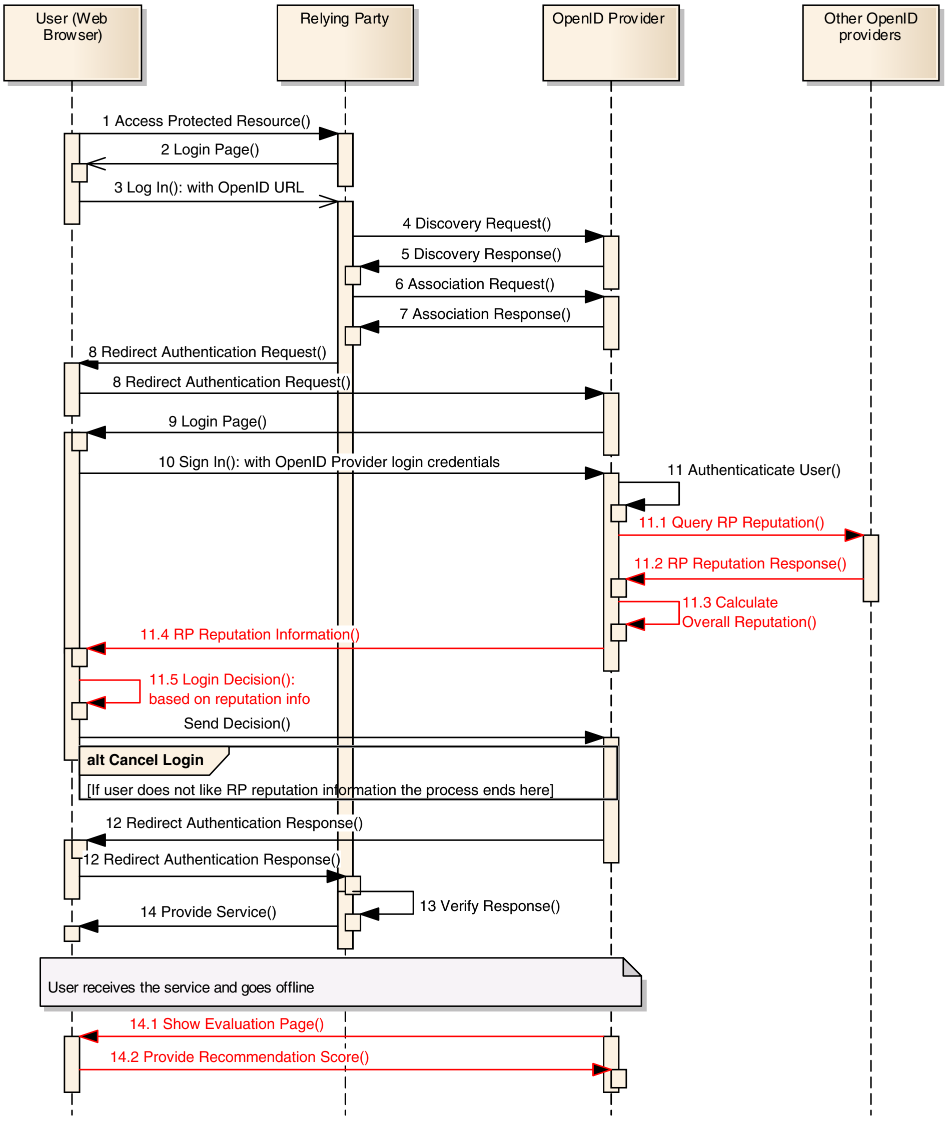

User (Web Browser)
Relying Party
OpenID Provider
Other OpenID providers
1 Access Protected Resource()
2 Login Page()
3 Log In(): with OpenID URL
4 Discovery Request()
5 Discovery Response()
6 Association Request()
7 Association Response()
8 Redirect Authentication Request()
8 Redirect Authentication Request()
9 Login Page()
10 Sign In(): with OpenID Provider login credentials
11 Authenticaticate User()
11.1 Query RP Reputation()
11.2 RP Reputation Response()
11.3 Calculate Overall Reputation()
11.4 RP Reputation Information()
11.5 Login Decision(): based on reputation info
Send Decision()
alt Cancel Login
[If user does not like RP reputation information the process ends here]
12 Redirect Authentication Response()
12 Redirect Authentication Response()
13 Verify Response()
14 Provide Service()
User receives the service and goes offline
14.1 Show Evaluation Page()
14.2 Provide Recommendation Score()

Fig. 2. OpenID protocol enhancement: sequence diagram.

Furthermore, this decision could be taken automatically by the OpenID provider based on pre-configured setting made by the users. For instance, the users may establish some rules in their OpenID provider, indicating that if the reputation of the RP has a score of at least 4 out of 5, the attributes will be released without explicit consent.

In case the user decides to continue with the process, the service is actually provided. Eventually, the user has the opportunity to evaluate the RP and provide her feedback to the OP (steps 14.1 and 14.2). The OpenID provider offers some mechanism where the users can rate the service or even provide additional comments. For instance, The OP would show a section in its website showing the pending-evaluation services once they login in the future, or send an email to them with a web page link where they can evaluate the received services.

Such information will be, in turn, used by forthcoming users to keep the most updated reputation value reflecting the current behavior of the RP. Furthermore, the users should be able to modify their recommendations afterwards in case they discover misleading behavior of the service, such as the RP starts to send spam, the conditions of the service change, or it is discovered that the RP reveals private information to third parties.

6. Reputation framework

This section shows the reputation framework designed for enhancing the OpenID users' experience when accessing a relying party. It will describe each one of the components [45], features and processes constituting the whole architecture.

6.1. Gathering recommendations

In order to determine how trustworthy an entity is, fulfilling the majority rating evaluation requirement (see Section 4), the first step of any trust and reputation infrastructure is collecting behavioral information about certain entity, in this case, the relying party. To collect as much information as possible, that information (i.e., recommendations) might come from different sources. In this proposal, the recommendations not only come from direct experiences the OP has had with a given RP, but also different recommendations are collected from other OpenID providers.

This enables obtaining accurate reputation values even though the OP does not have enough amount of registered users to obtain recommendations. Furthermore, although external recommendations are being requested, information about the users who provide such recommendations is not released, fulfilling the previously presented privacy requirement.

The first issue to solve when the specified OP wants to compute the reputation of the selected RP is to find those other OpenID providers that might have information about that concrete RP. To this end, we have designed the next subscription/notification mechanism.

6.1.1. Dynamic publish/subscribe mechanism

As soon as one of the end users of an OP wants to access a certain RP for the first time (the OP has never had any transaction with such RP in the past), then the OP sends a subscription request to that RP. Every RP keeps a list with the most recent OPs that have had an interaction with each of them (and therefore might have recommendations/opinions to provide about such RPs).

Thus, the RP will notify all the OP providers subscribed to it when this list of potential recommenders is updated. This publish/subscribe mechanism makes use of secure communication channels which require previous authentication of the parties to avoid any attacker to inject misleading information.

Additionally, in order to avoid an excessive flooding and overhead, such notification will take place with a certain frequency. Moreover, this frequency will dynamically change throughout the time in order to avoid unnecessarily flooding the system with non-needed messages, while keeping subscribers updated when such information is really necessary.

Hence, this list of OP providers will be sent to the subscribed OPs only when it contains Δ (Delta) new entries, whereas the actual value of Δ will determine the real frequency of the notifications. Thus for instance a value of $\Delta = 10$ would mean that such list of OPs would not be sent to the subscribers until 10 new entries are inserted in the list.

In order to dynamically adapt such value, every time a user accesses a certain RP through her OP, the value of Δ would decrease, increasing this way the frequency of notifications, since more users are interested in such RP and therefore the OP needs to have the most up-to-date information as possible. However, after certain time without receiving requests, the associated Δ would increase (decreasing the frequency of notifications), since the users of the OP are less interested in the services of the RP and the OP does not need to be continuously updated with the latest sources of recommendations for such RP.

Additionally, Δ would be bounded by a minimum value (to avoid an excessively high frequency of notifications). On the other hand, it should also have a maximum value. This value, when reached, should cause the OP to remove the subscription to that RP, since any of the users of such OP is no longer interested in such RP.

6.1.2. User-tailored recommendations

In order to accomplish with the users' preference awareness requirement described in Section 4 and to provide customized and user-tailored reputation information, each query for recommendations issued by the OP comes with the preferences of the end-user related to the provision of the final service (with regards to price, quality of service, delivery time, etc), although without revealing the identity of the user.

Thus, for a given end-user ($user_i$) a higher weight $\left(\omega_{user_i,user_j}\right)$ will be given to those recommendations $\left(Rec_{user_j}(RP)\right)$ coming from a user ($user_j$) whose service preferences $\left(Pref_{user_j}\right)$ match with the end-user ones $\left(Pref_{user_i}\right)$, since both share predilections or priorities and therefore the opinions of the former might be very valuable for the latter.

$$\omega_{user_i,user_j} = f_1\left(Pref_{user_i}, Pref_{user_j}\right)$$

6.1.3. Weighting aggregated recommendations

As to fulfill the incorrectness awareness requirement (see Section 4), when an OpenID is aggregating recommendations about a RP, it establishes a weight to each source of recommendations, both users and other OpenID providers. This weight (ω_{OP_i} for external OPs and ω_{user_i} for users) represents how much reliable the information given by the recommender is. Depending on this weight factor, the OpenID providers can treat information provided more or less relevant for the overall calculation of the relying party reputation.

This weight factor should be dynamically adapted based on the difference between the recommendations given by a specific recommender and those given by the rest of recommenders. In general, the closer the recommendations given by a specific source are to the average recommendations, the more unlike to be such source biased. Hence, the weight of a given OpenID provider OP_i, or a given user $user_i$ could be adjusted when a direct recommendation (e.g. given by $user_j$) is received.

$$\omega_{OP_i} = f_2\left(Rec_{user_j}, Rec_{OP_i}(RP)\right)$$

$$\omega_{user_i} = f_2\left(Rec_{user_j}, Rec_{user_i}(RP)\right)$$

6.1.4. Forgetting factor

Time awareness requirement shown in Section 4 entails assigning a higher weight to most recent transactions (and, consequently, their corresponding users' recommendations), in contrast to older ones, which might be considered less important. Thus, we are able to more accurately predict the actual current behavior of the given RP. Therefore, each recommendation $\left(Rec_{user_j}(RP)\right)$ is additionally given a weight $\left(\omega_{t,Rec_{user_j}(RP)}\right)$ which is obtained as follows:

$$\omega_{t,Rec_{user_j}(RP)} = f_3\left(t, time\left(Rec_{user_j}(RP)\right)\right)$$

where t is the current instant of time, while $time(Rec)$ is a function returning the time when recommendation Rec was provided.

6.2. Dynamically interchangeable reputation computation engine

So once the designed OP receives all the recommendation information from other OpenID providers (step 9.2 in Fig. 2), it has to aggregate it properly in order to compute the final reputation value for a given user of the relying party at a specific time $Reputation_{user_i,t}$ (RP).

This reputation computation component should take several elements into account when calculating such score, namely: the recommendations of their user and other end-users belonging to other OpenID providers $\left(Rec_{user_j}(RP)\right)$, the weight given to each of those recommendations based on the matching of users' preferences ($\omega_{user_i,user_j}$), the weight associated to each OP, measuring the reliability of its recommendations (ω_{OP_i}) and the so-called forgetting factor $\left(\omega_{t,Rec_{user_j}(RP)}\right)$.

The reputation calculation engine of our reputation framework should be designed in such a way that it supports multiple reputation computational models. Different example computation engines are presented in Section 7. Those computation models should be exchanged

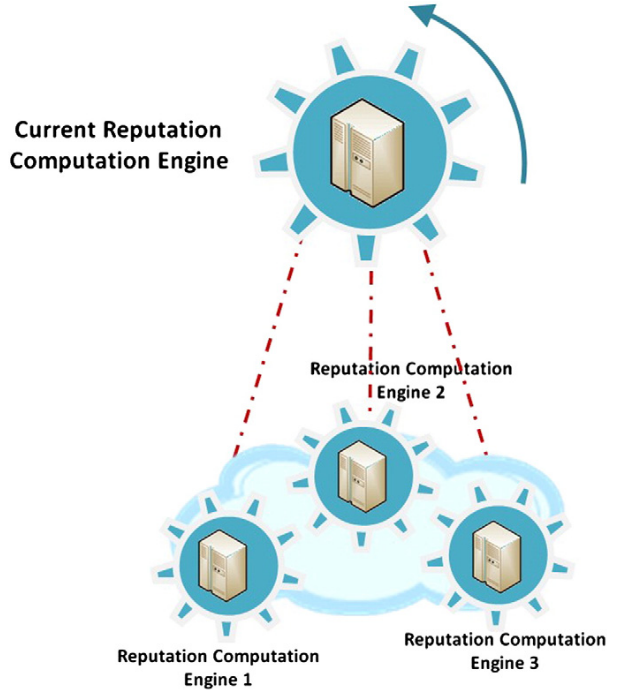

Current Reputation Computation Engine
Reputation Computation Engine 2
Reputation Computation Engine 1
Reputation Computation Engine 3

Fig. 3. Dynamically interchangeable reputation computation engine.

easily so that the framework can adapt to different scenarios on the fly, based on current conditions or circumstances (computation or network resources, storage resources, number of feedbacks, etc.), as shown in Fig. 3.

The framework would therefore seamlessly select the optimal reputation computation engine depending on the current conditions of the system, with the aim of adapting to those dynamic circumstances and to provide the user with the more accurate reputation scores at every time, without degrading the performance of the system or the user's experience.

6.3. General overview

As a summary, next we present the steps to be followed by our proposal, as depicted in Fig. 4.

1 Alice wants to watch a film at RP1.
2 Alice is redirected to OP1 in order to log-in and therefore share her Open ID with RP1.
  2.1 If OP1 is not subscribed to RP1, OP1 sends a subscription request to RP1.
2.1.1 RP1 replies with the list of OPs that have interacted with RP1.
  2.2 If OP1 was already subscribed to RP1, then RP1 decreases the value of Δ associated to OP1.
3 OP1 has the list of other OPs that have interacted with RP1 [13] (either because it was previously subscribed and already got it, or because it obtained it in step 2.(a).i). Therefore, OP1 sends a request to each of those OPs, asking for their respective recommendations about RP1. It also sends the preferences of the end-user (Alice in this example) preserving her privacy.
4 Each queried OP replies with a tailored recommendation based on the received preferences of the end-user.
5 OP1 collects and aggregates all the received recommendations.
6 OP1 applies the selected reputation computation mechanism and provides a final reputation score about RP1 to Alice.
7 Alice then decides, based on such reputation value, whether to trust the RP1 and go on with the process, or finish/cancel here the whole transaction.

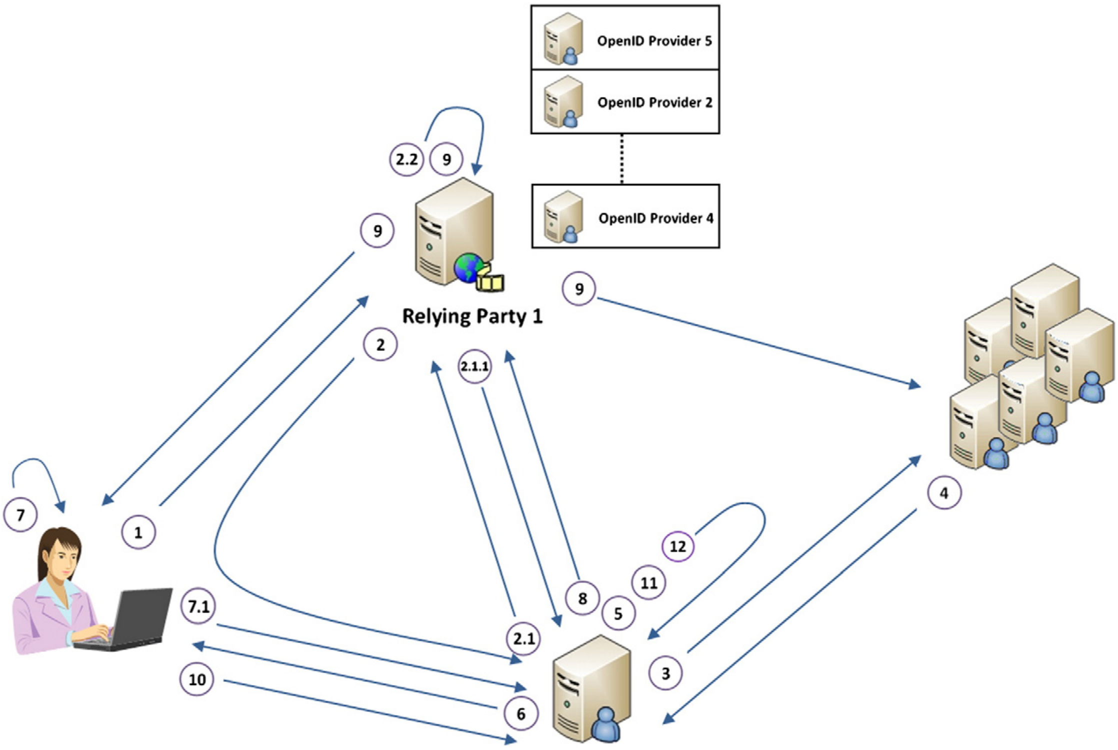


Fig. 4. General overview of the reputation framework.

7.1 Alice notifies OP1 about her decision, giving explicit consent to release the requested attributes if she has decided to continue.
8 If Alice trusts the RP1, then her profile is shared and sent from OP1 to RP1, where she is now logged-in.
9 RP1 provides the service to Alice, together with an electronic receipt. The RP1 also updates his list of recommenders, including OP1. If applicable, according to the current value of Δ, the OpenID providers subscribed to RP1 are notified with the updated list of recommenders.
10 Alice assesses her satisfaction with the received service and provides a recommendation about RP1 in her OP1, presenting the electronic receipt obtained in step 9.
11 OP1 updates its database of recommendations about RP1.
12 OP1 updates its reliability weights associated to other OPs.

7. Reputation computation engines

The main objective of the presented reputation framework is to collect and aggregate recommendations from different sources, in order to give additional information to the user about a specific relying party, before accessing its services. As introduced in the previous section, there are different ways to aggregate the collected recommendations, in order to compute the reputation value for a given relying party. These several mechanisms are implemented by the OpenID providers as computation engines, in such a way that each OpenID provider can dynamically choose the engine for calculating the reputation values depending on the system conditions, as shown in Fig. 3.

The results of each computation engine depend on the aspects or elements taken into account when performing the calculation, like users' preferences, among many others. Furthermore, the computation engines not only differ on the way of performing the reputation calculation but also in the resources they need to work. The aim of this section is to introduce different computation engines in order to analyze the feasibility of each of them under certain system conditions.

7.1. Average

The first and most straightforward computation engine that we describe is named Average. Being the recommendations provided by users or OpenID providers a real number belonging to the same interval ([0,1], for instance), this engine computes the reputation value for each relying party as an arithmetic mean in the following way.

On the one hand it calculates the arithmetic mean of all the available user recommendations. In case a user provides more than one recommendation about the same relying party, they all have to be taken into account to compute the reputation, but giving more weight to more recent ones. However, in order to avoid having to store all the recommendations given by a user, they are aggregated as soon as they are received by the OpenID provider.

When a user supplies an actual recommendation (in the moment t) the weight given to that recommendation depends on the time passed (Δt) since the last aggregation was calculated $\left(Rec_{user_i,t-\Delta t}\right)$, as shown in Eq. (1). Notice that Δt is a number between 0 and t ($\Delta t \in [0,t]$). Hence if Δt is nearly 0 (i.e. the two recommendations have been given closely in terms of time), they both will be almost equally taken into account, whereas if Δt is nearly t (i.e. plenty of time has passed between the two recommendations) almost only the new recommendation is considered. In this way, each user has just one recommendation value representing all given recommendations about a relying party, but giving a higher importance to more recent recommendations.

$$newRec_{user_i,t} = \left(\frac{1}{2} - \frac{\Delta t}{2t}\right) Rec_{user_i,t-\Delta t} + \left(\frac{1}{2} + \frac{\Delta t}{2t}\right) Rec_{user_i,t} \qquad (1)$$

On the other hand, it calculates the arithmetic mean taking all the previous OpenID provider recommendations about the specific relying party in the same ways as in Eq. (1) too. Finally, it aggregates these two arithmetic means giving a weight to each one (α and β, respectively), as represented in Eq. (2). Such weights, given to the users' recommendations and to the OpenID providers' recommendations respectively, could be adjusted beforehand, for instance by the system administrator, based on the scenario characteristics or other systems conditions.

$$Reputation = \alpha \left(\frac{1}{n} \sum_{i=1}^{n} Rec_{user_i} \right) + \beta \left(\frac{1}{m} \sum_{i=1}^{m} Rec_{OP_i} \right) \qquad (2)$$

α	weight of the users' recommendations
β	weight of the OpenID providers' recommendations
n	number of users' recommendations
m	number of OpenID providers' recommendations
Rec_{user_i}	aggregation of recommendations given by the ith user
Rec_{OP_i}	last recommendation given by the ith OpenID provider

7.2. Weighted average

This computation engine extends the previous one by assigning a weight to each user and OpenID provider, so the reputation values are computed performing a weighted average accordingly. The weights are defined with regards to the estimated goodness of each user or OpenID provider, in order to associate a lower importance to the recommendations given by malicious users or malicious OpenID providers when computing reputation values.

The goodness (i.e. the associated weight) of each user and each OpenID provider is calculated from the deviation of its recommendations compared to the recommendations of the rest of users and OpenID providers. For instance, a user will decrease her goodness if she provides low value recommendations while the rest of users provide high value recommendations to the same service, or vice versa. To do so, an initial weight is given to each user and OpenID provider, and they are updated when a user provides a new recommendation.

The weight of the user i at time t $\left(\omega_{user_i,t}\right)$ is calculated following Eq. (3).

$$\omega_{user_i,t} = f\left(\omega_{user_i,t-1}, \left|Reputation - Rec_{user_i}\right|\right) \qquad (3)$$

While the weight of the OpenID provider i at time t $(\omega_{OP_i,t})$ is calculated by Eq. (4).

$$\omega_{OP_i,t} = f\left(\omega_{OP_i,t-1}, \left|Reputation - Rec_{OP_i}\right|\right) \qquad (4)$$

Finally, the reputation value at time t is computed as a weighted average given by the formula presented in Eq. (5).

$$Reputation_t = \alpha \left(\frac{\sum_{i=1}^{n} \omega_{user_i,t} \cdot Rec_{user_i}}{\sum_{i=1}^{n} \omega_{user_i,t}} \right) + \beta \left(\frac{\sum_{i=1}^{m} \omega_{OP_i,t} \cdot Rec_{OP_i}}{\sum_{i=1}^{m} \omega_{OP_i,t}} \right) \qquad (5)$$

7.3. Preferences weighted average

The computation engines shown so far calculate global reputation values representing the opinion of all users as a unique value. However, they do not take into account that each user could have a different expectation about the same service, evaluating it with different values. In order to provide customized reputation information adapted to each user, OpenID providers should be able to compute user-tailored values depending on the similarity between users' preferences or profiles.

In order to measure such similarity, this computation engine takes into account the user preferences. These preferences express the assessment of each user with regards to the properties describing the service. Hence, this computation engine gives a higher weight to those recommendations coming from those other users whose preferences match with the user ones (the user who is currently accessing the service). The preferences assessment could be done when users are registered in the system, or the first time the user tried to access a kind of service. For instance, for a streaming video server the user could establish her predilection about parameters like video quality, audio quality, and price.

The Preferences weighted average computation engine establishes weights based on the similarity of the user preferences when calculating the reputation for a specific user. $pref_{user_n}$ being the set of preferences of the user n, this engine defines the similarity of user i with regards to user j ($sim_{i,j}$) as the deviation of their preferences over a given service parameters, as shown in Eq. (6).

$$sim_{i,j} = \sigma\left(pref_{user_i}, pref_{user_j}\right) \tag{6}$$

Thus, the reputation for the user j at time t is given by the formula described in Eq. (7).

$$\begin{aligned} Reputation_{user_j,t} = \alpha&\left(\frac{\sum_{i=1}^{n}\omega_{user_i,t}\cdot sim_{i,j}\cdot Rec_{user_i}}{\sum_{i=1}^{n}\omega_{user_i,t}\cdot sim_{i,j}}\right)\\ +\beta&\left(\frac{\sum_{i=1}^{m}\omega_{OP_i,t}\cdot Rec_{OP_i}}{\sum_{i=1}^{m}\omega_{OP_i,t}}\right) \end{aligned} \tag{7}$$

In turn, the recommendations collected by the OpenID provider from other OpenID providers, before performing the aggregation, are also customized for the specific user preferences.

7.4. Users weighted average

The previous computation engine calculates a customized reputation value assuming that all users having similar preferences evaluate the same service in a similar way. Additionally, it also assigns a global weight to each user and each OpenID provider estimating her goodness in order to avoid malicious actions.

The Users weighted average engine goes one step further and, instead of assigning just a weight, it establishes one weight for each pair of users and another one for each pair of user-OpenID provider. In this sense, when computing the reputation value for a specific user, this engine gives higher weight to those recommendations coming from users and OpenID providers whose previous recommendations were more similar to those given by the user [46].

The weight associated to the user i for the recommendations of the user j at time t is given by Eq. (8).

$$\omega_{user_i,user_j,t} = f\left(\omega_{user_i,user_j,t-1}, \left|Rec_{user_i} - Rec_{user_j}\right|\right) \tag{8}$$

While the weight associated to the user j for the recommendations of the OpenID provider i at time t is given by Eq. (9).

$$\omega_{OP_i,user_j,t} = f\left(\omega_{OP_i,user_j,t-1}, \left|Rec_{user_j} - Rec_{OP_i}\right|\right) \tag{9}$$

In this way, the reputation for the user i at time t could be represented as shown in Eq. (10).

$$\begin{aligned} Reputation_{user_j,t} = \alpha&\left(\frac{\sum_{i=1}^{n}\omega_{user_i,user_j,t}\cdot Rec_{user_i}}{\sum_{i=1}^{n}\omega_{user_i,user_j,t}}\right)\\ +\beta&\left(\frac{\sum_{i=1}^{m}\omega_{OP_i,user_j,t}\cdot Rec_{OP_i}}{\sum_{i=1}^{m}\omega_{OP_i,user_j,t}}\right) \end{aligned} \tag{10}$$

8. Experiments and results

This section describes the conducted experiments showing relevant aspects of the system behavior. The results obtained have been analyzed in order to validate the feasibility of the proposed framework. Finally, we summarize the main differences between the presented computation engines.

First of all, it is necessary to investigate the feasibility of the presented reputation solution. One of the main concerns when the framework was evaluated was the capability of malicious relying parties to exploit the reputation system. The point to explore is whether the relying party could easily increase its reputation score, or if it could be done by malicious users or OpenID providers. From this perspective, we have also analyzed the behavior of the different reputation computation engines in different scenarios.

8.1. Simulation settings

In order to analyze and compare the presented computation engines we have defined a simulation environment, representing the behavior of the reputation framework under different conditions. This section first describes the elements simulated to test and analyze the reputation framework, and secondly it characterizes the parameters used to compare the performance and requirements of different reputation engines.

8.1.1. Simulation descriptions

In order to evaluate the system behavior against different conditions, we have developed a tool able to simulate the actors and entities involved in the system. This tool is able to define virtual scenarios by specifying the amount of users, and the OpenID providers where they belong, and simulate their interactions with a given relying party.

In a scenario, different kind of users, OpenID providers and relying party could be defined, which allow analyzing the resilience to biased recommendations when the reputation is computed. For example, to determine the accuracy of the reputation framework if 20% of the users are providing biased recommendations, or if some of the OpenID providers are not honestly following the protocol. The different kind of actors and entities defined is described below. Furthermore, it could be specified which reputation computation engine the OpenID providers make use of in order to compare their output.

Once the scenario is defined, the developed tool generates random interactions between the different elements, following the previously defined protocol (see Section 6.3). To that end, a predefined number of simulation steps are executed, each of them consisting of a random subset of simulated users asking to their OpenID providers for the reputation value of the relying party, and interacting with it accordingly.

Fig. 5 depicts the phases executed in each simulation step. When a user requests a reputation value, her OpenID provider collects other OpenID providers and users recommendations, using the publish/subscribe mechanism described in Section 6.1.1, and aggregates them by making use of a specific computation engine.

The computation engines, which were developed within the tool, consist of the implementation of the equations described in Section 7. For example, the implementation of the Average computation engine

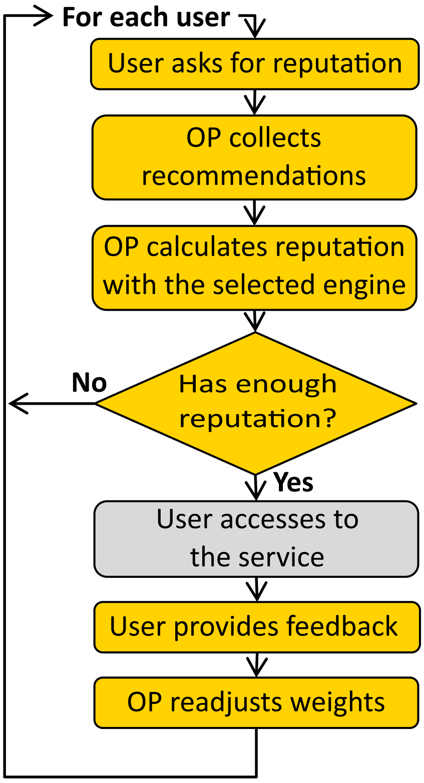


Fig. 5. Simulator execution phases.

computes and aggregates the arithmetic mean of the recommendations given by the users and those given by other OpenID providers (see Eq. (2)), assigning the same value to the weights of both the users' recommendations and the OpenID providers' recommendations ($\alpha = \beta = \frac{1}{2}$).

To compute the similarity function of the Preferences weighted average engine (see Eq. (6)), we have defined 5 aspects (representing for instance price, graphical interface, and usability) and each of them is rated by each user according to their preferences from 0, meaning not important at all, to 5, meaning that this aspect is very important to her. In this sense, the similarity between two users is computed based on the deviation of these preferences values.

Depending on the reputation value given to the user, she accesses (or not) to the relying party service. In the simulation tool, the users will interact with the relying party with a probability $p \in [0,1]$, p being the reputation value given to the relying party by the OpenID provider each user belongs to. Since this is a simulation environment, there is not a real service offered by the relying party, however this phase, where the user accesses the service, has been added in the figure to represent the real behavior of the system. After the user accesses the relying party, she provides her feedback to the OpenID provider, giving her opinion about how the service was. Finally, the OpenID provider, after receiving the feedback, recalculates the weights of the OpenID providers and users in case the calculation engine needs to do it, using the equations defined in Section 7.

For instance, the weights of the users in the Users weighted average computation engines are adjusted firstly computing the deviation between the feedbacks received for each pair of users, and then decreasing or incrementing their associated weights according to that deviation.

At the end of each simulation step, the tool logs the details of reputation framework, such as the average reputation given by each OpenID provider, the reputation received by each user, and the real quality received from the relying party. Executing several steps, the simulation gives information enough to analyze the behavior of the reputation framework against the defined scenario.

Moreover, simulations take into account possible attacks which could make the system vulnerable [39]. In order to define the elements taking part in the system, some premises regarding the relying party have been considered:

- The quality of service offered by a given relying party could vary during the time. For instance, a relying party providing good quality of service (therefore having a good reputation) could suddenly change its behavior and start giving a bad quality of service.
- A given relying party could influence the reputation score that each OpenID provider calculates about it. To this end, when an OpenID provider asks the relying party for the list of other OpenID providers having updated recommendations about such relying party, the relying party could decide to answer only with the OpenID providers which recommend it with high score.
- A given relying party could decide not to participate in the reputation framework. Hence, when an OpenID provider asks it for the list of OpenID providers which have had interaction with this relying party, it does not provide anything. It could take place either if the relying party prefers not to participate in order to avoid low score recommendations, or in case the relying party cannot implement the reputation framework, for instance due to lack of resources.

In the same sense, it should be taken into consideration that malicious users, or even malicious OpenID providers could be present in the system. With the term malicious we are referring, in this context, to those users or entities which try to degrade the reliability of the reputation framework. For instance, trying to ill-intentionally increase or decrease the reputation of a specific relying party.

Taking these considerations, we have defined different types of users, OpenID providers and relying parties acting in the system, as follows:

- Types of users:
  - Normal: These users provide appropriate recommendations according to the relying party quality of service.
  - Negative raters: This kind of user always provides low recommendation when giving the feedback, regardless of the relying party quality of service.
  - Positive raters: This kind of user always provides good recommendation values when giving the feedback, regardless of the relying party quality of service.
- Types of OpenID providers:
  - Normal raters: These OpenID providers represent the normal behavior within the system. That is, they collect user recommendations and other OpenID recommendations about the relying party, in order to calculate an accurate reputation value. When other OpenID providers ask them for the reputation value, they provide the calculated one.
  - Negative raters: Negative raters OpenID providers always give bad recommendation about the relying party (regardless the quality of service) trying to decrease the relying party global reputation.
  - Positive raters: Positive raters OpenID providers always give good recommendation about the relying party (regardless the quality of service) trying to increase the relying party global reputation.
  - Camouflaged positive raters: This kind of OpenID provider is an extension of the previous one. They give good recommendation, regardless the real quality of service, but only *p*% of the times. The rest of times, (*100-p*)%, they have a normal behavior.
  - Sybil positive raters: Sybil positive raters OpenID providers always provide good recommendation (regardless the quality of service), but additionally, after a while they are disconnected

and replaced with a new identity. These providers try to perform the Sybil attack [47], where a single malicious entity can present multiple identities, issuing a substantial fraction of the recommendations of the system.

- Types of relying party:
  - Normal: The relying party acts normal. When an OpenID provider asks for the list of OpenID providers with whom it recently interacted, it gives the real list.
  - Malicious: The relying party, when queried to get the OpenID providers with whom it recently interacted, it always delivers a list containing the ones with better recommendations about itself.
  - Sybil: The relying party, after a while, is disconnected and replaced with a new identity. This allows the relying party to reset its associated reputation, since it will be seen as a new entity.
  - Not participative: The relying party does not return the OpenID providers with whom it has recently interacted.

These actors and entities interact with each other simulating different environments where the reputation system could be deployed. The purpose of the simulations is to know if the users applying this solution will be properly informed about the quality of the requested services before accessing them. That is, if the reputation framework is able to compute a correct and accurate reputation value despite the aforementioned malicious elements.

8.1.2. System conditions and performance measurements

Once we have presented the different types of users, OpenID providers and relying parties that we are considering for our simulations, we will analyze different system or environment conditions and study their influence in each of the reputation engines described in Section 7. Furthermore, we will apply several performance measurements in order to assess the reliability of each of those computation engines. Such study will allow us to identify which is the most suitable engine for each situation, depending on the scenario requirements (more accuracy, more resilience, etc), and the current system conditions (network resources, number of collected feedbacks, etc).

Next we present several parameters describing the system conditions that could mainly influence each of the considered reputation engines. Simulation outcomes shown later will determine the adaptability of each engine with regards to each system condition.

- Number of users: This parameter represents the amount of end users participating in the system.
- Number of OpenID providers: It represents the amount of OpenID providers participating in the system.
- User participation: This parameter specifies how participative the users are within the system. In other words, it indicates whether the users participating in the system are being active and continually requesting services to make the reputation framework efficiently work.
- Network resources: It indicates how many network resources (in terms of bandwidth, for instance) are present in the system.
- Computer resources: It indicates how many computer resources (in terms of computation capacity, storage, etc.) every OpenID provider has, on average.

In order to study the feasibility of each reputation engine, we have defined several performance measurements as described next:

- Accuracy: This measurement indicates how similar the computed reputation score with regards to the actual goodness or behavior of the corresponding relying party is.
- User satisfaction: This measurement indicates the similarity between the reputation score provided by the framework to a user (regarding a concrete relying party) and the actual satisfaction or feedback of that user (with that specific relying party).
- Adaptability: This measurement indicates the ability of the reputation engine to quickly and accurately react to sudden changes in the behavior of the relying party, by recalculating an appropriate new reputation score.
- Behavior with malicious users: This measurement indicates the level of resilience of the analyzed engine with regards to malicious users.
- Behavior with malicious OpenID providers: This measurement indicates the level of resilience of the analyzed engine with regards to malicious OpenID providers.

8.2. System conditions

Several tests have been performed in order to analyze the presented computation engines against the system conditions presented in Section 8.1.2. This section summarizes the main tests performed presenting preliminary results about each of those system conditions.

8.2.1. Number of users

There are computation engines which need a large amount of users to achieve an optimal performance. On the other hand, there are computation engines whose behavior is not affected by the number of users acting in the system. In order to determine this condition, we have performed tests simulating different OpenID providers, including malicious ones, changing the amount of users belonging to each of these OpenID providers.

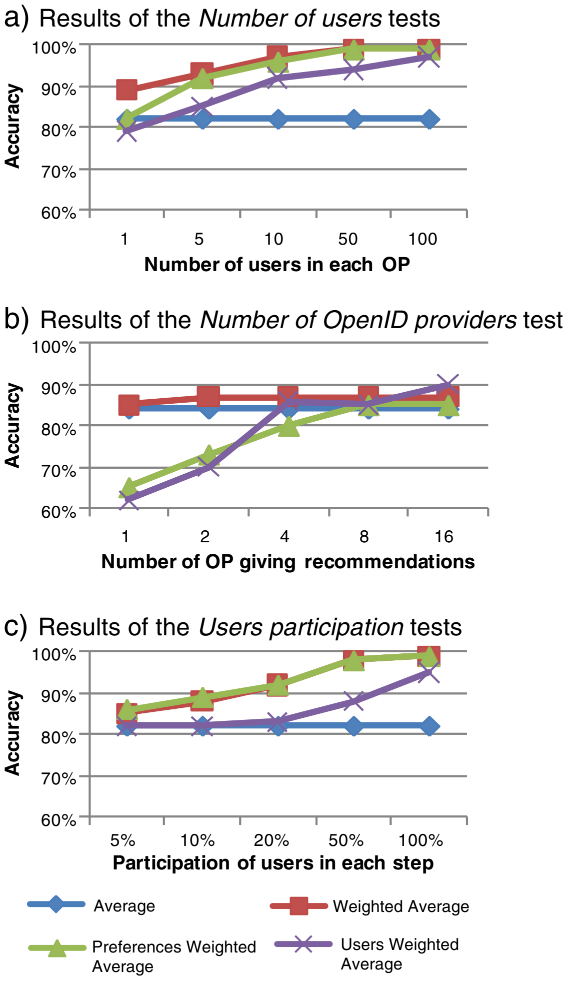


Fig. 6. Experiment outcomes analyzing some current system conditions.

In the tests deploying several malicious OpenID providers, trying to raise the reputation value of the relying party, we have realized that the more users are deployed in the system the more accurate some models are. A summary of the results of these experiments is shown in Fig. 6. They show that the Average computation engine does not depend on the number of users at all, but in other factors, such as the percentage of malicious users as we show later on. That is, the result of an arithmetic mean does not depend on the number of addends if all of them are equals. Both Weighted average and Preferences weighted average engines get better results when there are more users in the system, although they do not need a vast amount of users to get good results. That is why the weight given to a specific user is mainly based on the recommendations of the rest of the users. The more users participating, the more recommendations there are and, therefore, the more precise the weights are. Yet, the Users weighted average engine needs about 100 users per OpenID provider to achieve an accuracy of 97%. That is due to the fact that this model, for each user, needs other users with similar behavior to obtain an adapted reputation value.

8.2.2. Number of OpenID providers

Extending the previous point, we analyzed the number of OpenID providers that each computation engine needs in order to get an optimal performance. This time, we have simulated different scenarios containing all types of users, changing the amount of OpenID providers which provide recommendations about a given RP, but maintaining the amount of users in each of them.

For some calculation engines, the number of OpenID providers does not affect the accuracy of the system. However, there are other calculation engines where tests have shown that the accuracy and user satisfaction could grow when increasing the number of OpenID providers. However, when this number reaches an upper limit, then the behavior remains constant. That is, from a specific point the accuracy does not improve even if the number of OpenID providers increases. For instance, Fig. 7 shows the results of the user satisfaction, supposing that there are just 2 users supplying recommendations on each OpenID provider.

These results are in fact related with the results of the number of users in the system, in the sense that if the OpenID providers do not have enough users belonging to them, they cannot offer adapted reputation values, because they do not have enough recommendations. However, these OpenID providers could increase the number of recommendations asking other OpenID providers.

On the one hand, Average and Weighted average engines do not need too many OpenID providers to reach such upper limit, since the recommendations given by their users are usually enough. On the other hand, Preferences weighted average and Users weighted average engines need more recommendations to calculate accurate reputation values. Therefore, they work better if they ask other OpenID providers for recommendations when there are not enough users on each OpenID provider.

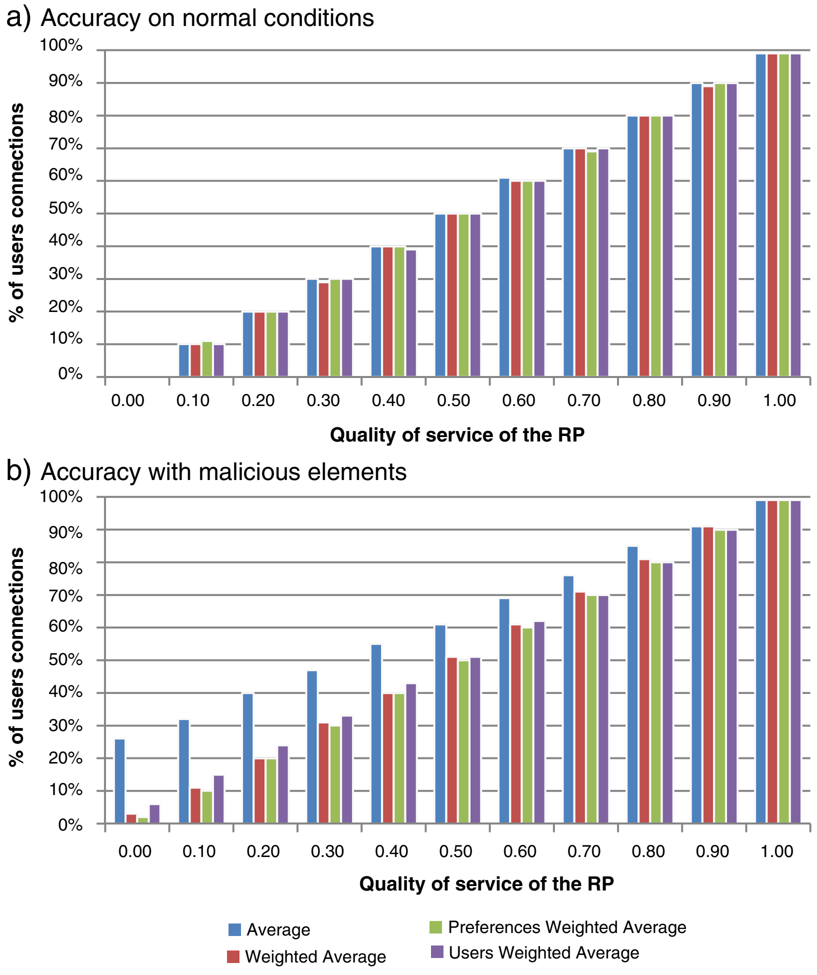


Fig. 7. Experiment outcomes analyzing accuracy of the reputation computation engines.

8.2.3. User participation

Besides the number of users and OpenID providers in the system, the accuracy of some models also depends on the participation of the users in the system. For example, even if there were a lot of users belonging to a specific OpenID provider, such OpenID provider might not be able to compute a reliable reputation value if its users are not regularly providing feedbacks.

In order to analyze this parameter, we have simulated similar scenarios than in the previous tests, but in this case modifying the frequency of participation of the users in the system, therefore providing more or less feedbacks. As shown in Fig. 8, in general terms, the frequency of users participation does not affect to the Average computation engine accuracy, since few recommendations are needed to reach an upper limit using this engine, as previously shown. Nevertheless, in other simulated scenarios where there are not many users or several malicious OpenID providers has been deployed, the rest of the computation engines, specially the Users weighted average one, could be linked to this parameter. Since these models have to adjust the weight related to each of its users, they require the users providing more recommendations to successfully adjust such weights.

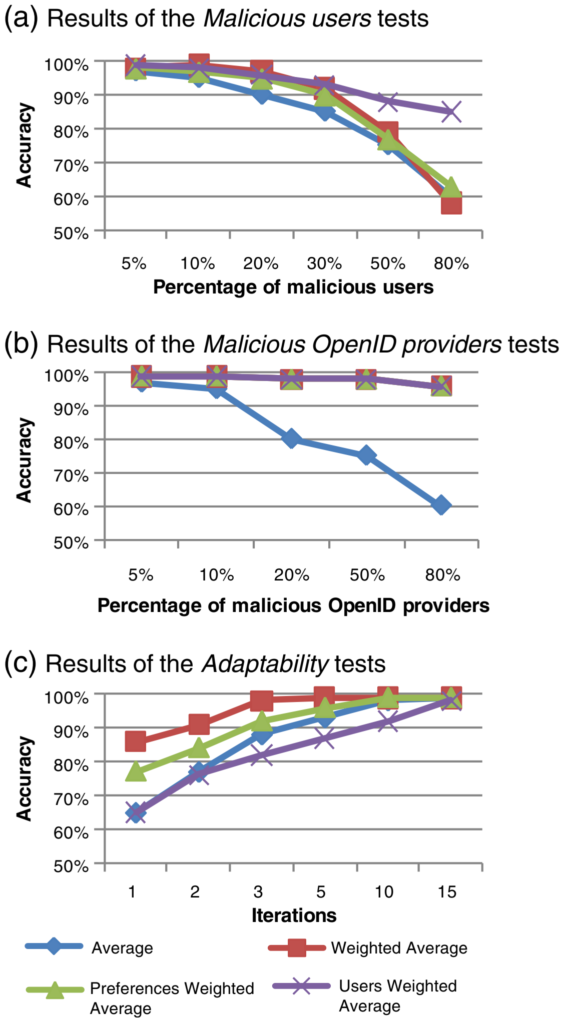


Fig. 8. Experiment outcomes analyzing some performance measurements.

8.2.4. Network resources

In order to collect external recommendations, the OpenID providers have to send queries to other OpenID providers, which in turn have to send the response back containing the recommendation. So then, the more recommendations a computation engine needs to collect the more network resources it requires.

Since the Average and Weighted average models do not adapt the reputation values to each user, they collect global recommendations. In this sense, once collected the external recommendations, they can be used to compute the reputation value of any of the users. They just need to ask an OpenID provider for recommendations again, when the previously collected recommendations are out of date.

On the other hand, the Preferences weighted average and the Users weighted average models should collect different recommendations for each user (or group of users). Therefore, the network resources that these computation engines require to work are higher.

8.2.5. Computer resources

All models need to store collected recommendations, both user feedbacks and other OpenID recommendations, in order to apply the computation algorithm to calculate the reputation. Furthermore, some calculation engines also make use of additional information to avoid malicious behaviors or to get adapted reputation values.

- The Average computation engine just needs to store and compute the users and OpenID provider recommendations.
- The Weighted average engine has to store the weight of each user and OpenID provider, which should be also adapted after receiving the user feedback.
- Additionally, the Preferences weighted average engine has to store the preferences of each user, which has to be also computed in order to determine the similarity between users.
- The Users weighted average engine stores a weight for each pair of users and each pair of user-OpenID provider which could require a vast amount of computer resources if there are lot of users deployed in the OpenID provider, although it could be avoided by making use of grouping techniques. In addition, it should compute a reputation value for each user.

Thus, the Average computation engine is the one needing fewer resources, among the studied engines.

8.3. Performance measurement

This section presents the results achieved in order to evaluate the performance measurements between the different computation engines described in Section 7.

8.3.1. Accuracy

All studied models obtain good accuracy measurements supposing nice conditions in the simulated environment, that is, without adding malicious components, as depicted in the Fig. 7a. Taking into account that the users will access the relying party with a probability given by the reputation value, this plot shows the percentage of users which has accessed the offered service when the relying party has a specific quality of service. Therefore, the percentage of user connections should be equal to the quality of service (expressed as a percentage) of the relying party if the model has optimal accuracy. For instance, if the relying

Table 1
Experiment outcomes analyzing user satisfaction on reputation computation engines.

Computation engine	User satisfaction	Standard deviation
Average	88%	11.4%
Weighted average	89%	11.6%
Preferences weighted average	96%	0.4%
Users weighted average	98%	0.3%

party has a quality of service of 30%, the closer the percentage of user connections to 30% is, the more accurate the model is.

However, when there are malicious components in the system the accuracy of the reputation values given by the OpenID providers could be reduced. As shown in Fig. 7b, the accuracy of the Average computation engine is reduced in scenarios containing malicious elements, since it does not implement any mechanism to evaluate the goodness of the collected recommendations. The rest of the computation engines calculate more accurate recommendation values even receiving unsuitable recommendations, since these engines are, in certain way, resilient to malicious users or OpenID providers, as we will discuss in the following sections.

8.3.2. User satisfaction

By performing some experiments under normal conditions, that is, without malicious OpenID providers or malicious users, we have demonstrated that the Preferences weighted average and Users weighted average engines provide significantly more adapted reputation values to the users. This is due to the fact that these computation engines, when calculating the reputation for a specific user, they associate a higher importance to the recommendations given by other users with similar behavior. A summary of the performed test is depicted in Table 1.

Average and weighted average computation engines, although they accurately calculate the average reputation value, they do not provide correct reputation value to those users whose opinions are far from the average.

8.3.3. Adaptability

If a relying party has good reputation but it suddenly starts to provide bad services, it should be quickly detected by the reputation system. Some tests have been performed in this direction in order to detect how long the different engines take to discover the new quality of service of the relying party. Fig. 8c shows general result of these tests, supposing that a relying party radically changes its quality of service. Considering an iteration the process where all users perform the execution phases, previously described, it indicates the accuracy of the computation engines calculating the reputation value after a given number of iterations.

The Average computation engine takes several iterations to find the updated quality of service, because during some iterations it still takes into account past recommendations related to the previous reputation value. The Weighted average punishes inaccurate recommendations, either given by malicious users or out of date, so it quickly adapts to the current reputation value.

Those computation engines which base the weights on the similarity between users take more time to adapt the reputation value to the current quality of service, specially the Users weighted average one. That is due to the fact that they should readjust all the weights after the quality of service changes, so they need a higher amount of feedbacks. In general, the more users deployed in the system the longer the computation engine takes to discover the new quality of service.

8.3.4. Behavior with malicious users

Although under nice conditions the presented computation engines have demonstrated to calculate accurate reputation values, they also have to be resilient against user attacks. By modifying the percentage of malicious users distributed throughout the system, we have performed different tests in order to analyze this parameter, whose outcomes are summarized in Fig. 8a.

The inaccuracy of the Average engine directly depends on the percentage of the malicious users, since it does not implement any mechanism to detect them. The Weighted average engine is, to some extent, resilient to malicious users, since it will provide less weight to those user opinions which are not similar to the rest of the user opinions. Additionally, the Preferences weighted average is slightly less resilient against malicious users, since the weights it provides are also depending on the users' preferences.

Finally, the Users weighted average engine is the most resilient to malicious users due to the fact that it tries to match users who are providing similar feedbacks. Therefore, the users providing inappropriate feedbacks will not be taken into account by the users which provide feedbacks according to the relying party quality of service.

8.3.5. Behavior with malicious OpenID providers

Similar to the previous section, we have investigated the behavior of the engines against malicious OpenID providers. A summary of testing different scenarios, modifying the percentage of OpenID providers, is shown in Fig. 8b.

Except for the Average computation engine, which does not implement any mechanism to detect malicious OpenID providers, the computation engines are resilient to malicious OpenID providers, in general. If the recommendations given by a specific OpenID provider are not related to the feedbacks provided by the users, the weight of such OpenID provider decreases notably, skipping its recommendations in future interactions.

8.4. Summary

Conducted experiments have proven the feasibility of the reputation framework, since the evaluated computation engines have shown a good performance, providing accurate reputation values and good user satisfaction, even in the case where several malicious entities were present in the system.

The reputation values are not determined by the honesty of the relying party, i.e., whether it is malicious or not. Instead, the reputation values could be conditioned by the number of malicious OpenID providers, although they could be avoided by the reputation computation engine. For instance, if there is no OpenID provider highly scoring a given malicious low-quality relying party, such relying party is not able to interact with any OpenID provider. In case there are OpenID providers highly scoring this concrete relying party, these OpenID providers could be marked as malicious by the computation engine. Therefore, their recommendation could not be taken into consideration.

Besides, one of the main conclusions reached after performing these simulations is that there is not an ideal computation engine which can be used in every environment. Instead, their performance depends on the current conditions present in the system. For instance, if we have to deploy the reputation framework in a system where just a few users are going to interact, or these users are not so active, we will probably have to make use of the Average or Weighted average computation engine, since the other engines need many feedbacks to acquire good accuracy in the reputation values.

Likewise, although the Preferences weighted average and the Users weighted average computation engines provide more adapted reputation values, they need more network and computer resources to be deployed, so in systems having limited resources they are not a good option. Table 2 summarizes a comparison between different computation engines regarding the system conditions.

Besides the system conditions, the choice of the computation engine should also depend on the specific performance demands of the application scenario. For instance, if an OpenID provider wanted to provide

Table 2
Reputation computation engines comparison analyzing some current system conditions.

Computation engine	Number of users	Number of OPs	User participation	Network resources	Computer resources
Average	Low	Low	Low	Low	Low
Weighted average	Medium	Low	Medium	Low	Medium
Preferences weighted average	High	Medium	Medium	High	High
Users weighted average	Very high	High	Very high	High	Very high

Table 3
Reputation computation engines comparison analyzing some performance measurement.

Computation engine	Accuracy	User satisfaction	Adaptability	Behavior with malicious users	Behavior with malicious OPs
Average	Medium	Poor	Medium	Poor	Poor
Weighted average	Good	Slightly poor	Good	Medium	Good
Preferences weighted average	Good	Good	Slightly good	Medium	Good
Users weighted average	Good	Good	Medium	Good	Good

adapted recommendation to their users, it should choose the Preferences weighted average or Users weighted average engine. However, if it needs an engine which quickly detects the changes of the relying party quality of service, it should select the Weighted average engine instead, although it has less user satisfaction. Table 3 summarizes the performance measurements outcomes.

Finally, it is worth mentioning that the OpenID provider should be able to dynamically choose the computation engine, selecting in each case the one giving more accurate reputation values, depending on the system conditions and the expected performance parameters. To this end, we are currently investigating a new and novel system capable of analyzing such system conditions on-the-fly, and to select the computation engine that better fits with the defined (expected) performance metrics.

9. Conclusions and future work

In the current Internet there are many service providers, most of them are being aimed to provide appropriate services while some others are not so well intended. In this context, it is interesting for any end user to have mechanisms to determine how trustworthy a particular service provider is, so she can decide if she wants to interact with it or not. It is particularly interesting if this service is requesting some of her personal information (email address, bank account, age, etc.) before granting access to any of the resources the service provider has.

This is a problem that should be addressed before starting the interaction with the system, i.e., before sending the end user attributes to the service provider. To this end, IdM solutions need to be adapted and enhanced with particular mechanisms enabling the provisioning of certain meta-information to the user on the particular service provider being accessed.

One of the SSO-enabled IdM solutions most widely developed nowadays is OpenID. A lot of service providers and certain key IdM providers are including this standard solution as part of the authentication and basic access control services provided to their end users. However, OpenID in its current definition can be used by a malicious service provider to gain access to the private attributes of users and make profit with them.

To provide end users with valuable behavioral information on the different service providers, this paper is defining a reputation framework and how it can be applied to an extended version of the OpenID protocol. In this way, this paper is describing a solution helping to mitigate this problem. It is based on the idea that users can provide a recommendation level on a particular service and it is later being aggregated by the OpenID provider and provided to any other potential user that might be interested to interact with the same service provider in the future. With this help, authors are provided with a mechanism aimed to increase their level of satisfaction with the OpenID system.

Performed experiments show that the reputation values are not determined by the honesty of the relying party. Furthermore, although the reputation values could be conditioned by the number of malicious users and OpenID providers, they can be avoided by certain reputation computation engines. We have analyzed different reputation computation engines, so the different pros and cons can be determined.

As for future work, since there is not a perfect reputation computation engine, that is, suitable for each conditions, we are developing a mechanism to perform a dynamic and automatic selection of the most convenient reputation computation engine at each moment based on the current system conditions and the specified performance measurements. Finally, such mechanism will comprise a smooth transition between computation engines.

Acknowledgment

This research work has been partially funded by the Next-Generation Hybrid Broadcast Broadband (HBB-NEXT) EU research project, FP7-ICT-2011.1.5, grant no.: 287848.

References

[1] J. Jensen, Benefits of federated identity management: a survey from an integrated operations viewpoint, Proceedings of the IFIP WG 8.4/8.9 International Cross Domain Conference on Availability, Reliability and Security for Business, Enterprise and Health Information Systems, ARES'11, Springer, 2011, pp. 1–12.

[2] D.W. Chadwick, Federated Identity Management, Foundations of Security Analysis and Design V, Lect. Notes Comput. Sci. 5705 (Springer 2009) 96–120.

[3] P. Arias Cabarcos, F. Almenárez Mendoza, A. Marín López, D. Díaz Sánchez, Enabling SAML for Dynamic Identity Federation Management, Wireless and Mobile Networking Conference, IFIP Advances in Information and Communication Technology, Springer, 2009, pp. 173–184.

[4] D. Smith, The Challenge of Federated Identity Management, Network Security, 2008.

[5] E. Maler, D. Reed, The venn of identity: options and issues in federated identity management, IEEE Sec. Priv. 6 (2) (2008) 16–23.

[6] D. Recordon, D. Reed, OpenID 2.0: a platform for user-centric identity management, in: Proceedings of the second ACM workshop on Digital identity management, DIM '06, 2006, pp. 11–16.

[7] H.-K. Oh, S.-H. Jin, The Security Limitations of SSO in OpenID, 10th International Conference on Advanced Communication Technology, volume 3, 2008, pp. 1608–1611.

[8] P. Patel, A. Ranabahu, A. Sheth, Service level agreement in cloud computing, in: Cloud Workshops at OOPSLA.

[9] M. Mostarda, D. Palmisano, F. Zani, S. Tripodi, Towards an openid-based solution to the social network interoperability problem, in: W3C workshop on the future of social networking, 2008, pp. 15–16.

[10] G. Dólera Tormo, G. López Millán, G. Martínez Pérez, Definition of an advanced identity management infrastructure, Int. J. Inf. Secur. 12 (2013) 173–200.

[11] A. Bhargav-Spantzel, A.C. Squicciarini, E. Bertino, Trust negotiation in identity management, IEEE Secur. Priv. 5 (2007) 55–63.

[12] D. Choi, S.-H. Jin, H. Yoon, S. Tripodi, Trust management for user-centric identity management on the internet, in: IEEE International Symposium on Consumer Electronics, 2007, pp. 1–4.

[13] F. Gómez Mármol, M. Kuhnen, G. Martínez Pérez, Enhancing OpenID through a reputation framework, Proceedings of the 8th International Conference on Autonomic and Trusted Computing, ATC'11, number 6906 in LNCS, Springer-Verlag, 2011, pp. 1–18.

[14] B.C. Neuman, T. Ts'o, Kerberos: an authentication service for computer networks, IEEE Commun. Mag. 32 (1994) 33–38.

[15] F. Pereñíguez GarcÃ-a, R. Marín-López, G. Kambourakis, A. Ruiz-Martínez, S. Gritzalis, A. Skarmeta-Gómez, KAMU: providing advanced user privacy in Kerberos multi-domain scenarios, Int. J. Inf. Secur. (2013) 1–21.

[16] J. Bobadilla, F. Ortega, A. Hernando, A. Gutiérrez, Recommender systems survey, Knowl.-Based Syst. 46 (2013) 109–132.

[17] F. Gómez Mármol, G. Martínez Pérez, Providing trust in wireless sensor networks using a bio-inspired technique, Telecommun. Syst. J. 46 (2011) 163–180.

[18] F. Gómez Mármol, J. Gómez Marín-Blázquez, G. Martínez Pérez, Linguistic Fuzzy Logic Enhancement of a Trust Mechanism for Distributed Networks, in: Proceedings of the Third IEEE International Symposium on Trust, 2010, pp. 838–845.

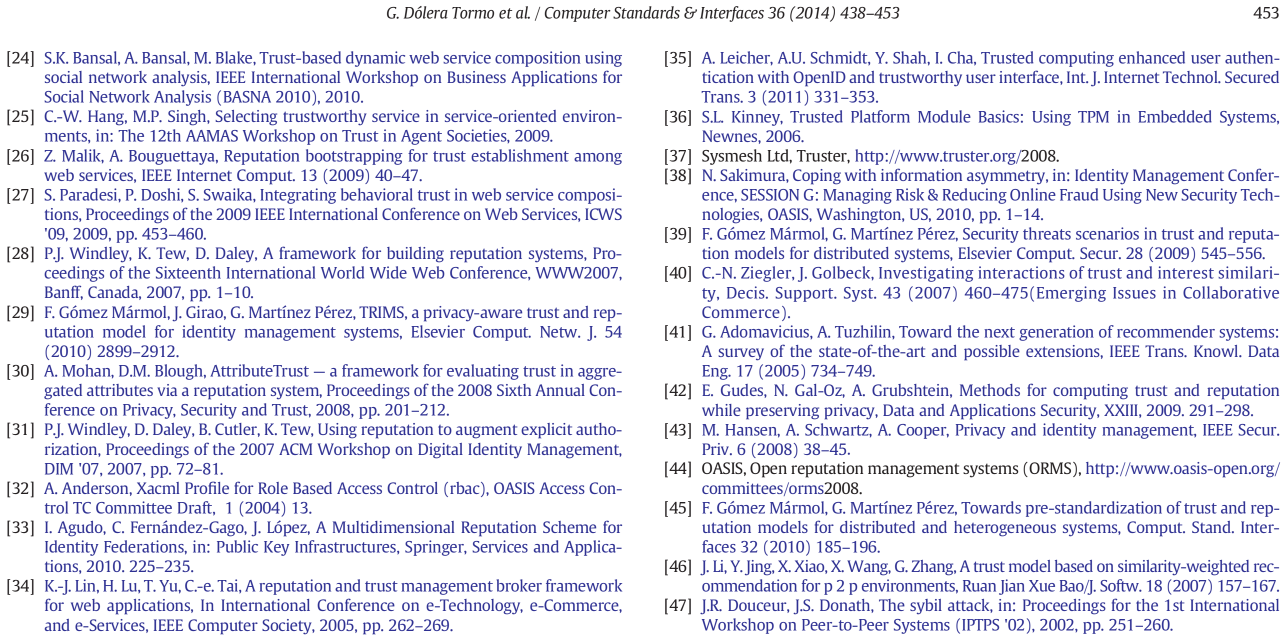
[19] M. Omar, Y. Challal, A. Bouabdallah, Reliable and fully distributed trust model for mobile ad hoc networks, Comput. Secur. 28 (2009) 199–214.

[20] F. Gómez Mármol, G. Martínez Pérez, A. F. Gómez Skarmeta, TACS, a Trust Model for P2P Networks, Wireless Personal Communications, Special Issue on "Information Security and data protection in Future Generation Communication and Networking" 51 (2009) 153–164.

[21] Y. Wang, Y. Tao, P. Yu, F. Xu, J. Lu, A Trust Evolution Model for P2P Networks, Autonomic and Trusted Computing, number 4610 in LNCS, 4th International Conference, ATC 2007, Springer, Hong Kong, China, 2007, pp. 216–225.

[22] C. Huang, H. Hu, Z. Wang, A dynamic trust model based on feedback control mechanism for P2P applications, Autonomic and Trusted Computing, number 4158 in LNCS, Springer, Wuhan, China, 2006, pp. 312–321.

[23] S. Marti, H. García-Molina, Identity crisis: anonymity vs reputation in P2P systems, Proceedings for the Third International Conference on Peer-to-Peer Computing (P2P 2003), IEEE Computer Society, Washington DC, USA, 2003, pp. 134–141, (Linköping, Sweden).



[24] S.K. Bansal, A. Bansal, M. Blake, Trust-based dynamic web service composition using social network analysis, IEEE International Workshop on Business Applications for Social Network Analysis (BASNA 2010), 2010.

[25] C.-W. Hang, M.P. Singh, Selecting trustworthy service in service-oriented environments, in: The 12th AAMAS Workshop on Trust in Agent Societies, 2009.

[26] Z. Malik, A. Bouguettaya, Reputation bootstrapping for trust establishment among web services, IEEE Internet Comput. 13 (2009) 40–47.

[27] S. Paradesi, P. Doshi, S. Swaika, Integrating behavioral trust in web service compositions, Proceedings of the 2009 IEEE International Conference on Web Services, ICWS '09, 2009, pp. 453–460.

[28] P.J. Windley, K. Tew, D. Daley, A framework for building reputation systems, Proceedings of the Sixteenth International World Wide Web Conference, WWW2007, Banff, Canada, 2007, pp. 1–10.

[29] F. Gómez Mármol, J. Girao, G. Martínez Pérez, TRIMS, a privacy-aware trust and reputation model for identity management systems, Elsevier Comput. Netw. J. 54 (2010) 2899–2912.

[30] A. Mohan, D.M. Blough, AttributeTrust — a framework for evaluating trust in aggregated attributes via a reputation system, Proceedings of the 2008 Sixth Annual Conference on Privacy, Security and Trust, 2008, pp. 201–212.

[31] P.J. Windley, D. Daley, B. Cutler, K. Tew, Using reputation to augment explicit authorization, Proceedings of the 2007 ACM Workshop on Digital Identity Management, DIM '07, 2007, pp. 72–81.

[32] A. Anderson, Xacml Profile for Role Based Access Control (rbac), OASIS Access Control TC Committee Draft, 1 (2004) 13.

[33] I. Agudo, C. Fernández-Gago, J. López, A Multidimensional Reputation Scheme for Identity Federations, in: Public Key Infrastructures, Springer, Services and Applications, 2010. 225–235.

[34] K.-J. Lin, H. Lu, T. Yu, C.-e. Tai, A reputation and trust management broker framework for web applications, In International Conference on e-Technology, e-Commerce, and e-Services, IEEE Computer Society, 2005, pp. 262–269.

[35] A. Leicher, A.U. Schmidt, Y. Shah, I. Cha, Trusted computing enhanced user authentication with OpenID and trustworthy user interface, Int. J. Internet Technol. Secured Trans. 3 (2011) 331–353.

[36] S.L. Kinney, Trusted Platform Module Basics: Using TPM in Embedded Systems, Newnes, 2006.

[37] Sysmesh Ltd, Truster, http://www.truster.org/2008.

[38] N. Sakimura, Coping with information asymmetry, in: Identity Management Conference, SESSION G: Managing Risk & Reducing Online Fraud Using New Security Technologies, OASIS, Washington, US, 2010, pp. 1–14.

[39] F. Gómez Mármol, G. Martínez Pérez, Security threats scenarios in trust and reputation models for distributed systems, Elsevier Comput. Secur. 28 (2009) 545–556.

[40] C.-N. Ziegler, J. Golbeck, Investigating interactions of trust and interest similarity, Decis. Support. Syst. 43 (2007) 460–475(Emerging Issues in Collaborative Commerce).

[41] G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions, IEEE Trans. Knowl. Data Eng. 17 (2005) 734–749.

[42] E. Gudes, N. Gal-Oz, A. Grubshtein, Methods for computing trust and reputation while preserving privacy, Data and Applications Security, XXIII, 2009. 291–298.

[43] M. Hansen, A. Schwartz, A. Cooper, Privacy and identity management, IEEE Secur. Priv. 6 (2008) 38–45.

[44] OASIS, Open reputation management systems (ORMS), http://www.oasis-open.org/committees/orms2008.

[45] F. Gómez Mármol, G. Martínez Pérez, Towards pre-standardization of trust and reputation models for distributed and heterogeneous systems, Comput. Stand. Interfaces 32 (2010) 185–196.

[46] J. Li, Y. Jing, X. Xiao, X. Wang, G. Zhang, A trust model based on similarity-weighted recommendation for p 2 p environments, Ruan Jian Xue Bao/J. Softw. 18 (2007) 157–167.

[47] J.R. Douceur, J.S. Donath, The sybil attack, in: Proceedings for the 1st International Workshop on Peer-to-Peer Systems (IPTPS '02), 2002, pp. 251–260.

4

ROMEO: ReputatiOn Model Enhancing OpenID Simulator

Title:	ROMEO: ReputatiOn Model Enhancing OpenID Simulator
Authors:	Ginés Dólera Tormo, Félix Gómez Mármol, Gregorio Martínez Pérez
Type:	International Conference
Conference:	19th European Symposium on Research in Computer Security (ESORICS), Security & Trust Management Workshop (STM 2014)
Publisher:	Springer, LNCS 8743
Pages:	193-197
Location:	Wroclaw, Poland
Year:	2014
Month:	September
DOI:	`http://dx.doi.org/10.1007/978-3-319-11851-2_15`
State:	Published

Table 4: ROMEO: ReputatiOn Model Enhancing OpenID Simulator

ROMEO: ReputatiOn Model Enhancing OpenID Simulator

Ginés Dólera Tormo[1], Félix Gómez Mármol[2], and Gregorio Martínez Pérez[1]

[1] Department of Information and Communications Engineering,
University of Murcia, Murcia, 30100 Spain
{ginesdt,gregorio}@um.es
[2] NEC Europe Ltd., Kurfürsten-Anlage 36, 69115 Heidelberg, Germany
felix.gomez-marmol@neclab.eu

Abstract. OpenID is a standard decentralized initiative aimed at allowing Internet users to use the same personal account to access different services. Since it does not rely on any central authority, it is hard for such users or other entities to validate the trust level of each other. Some research has been conducted to handle this issue, defining reputation framework to determine the trust level of a service based on past experiences. Deep analysis and validation need to be achieved in order to prove the feasibility of this framework. Our main contribution in this paper consists of a simulation environment able to validate the feasibility of that reputation framework and to analyze its behavior within different scenarios.

1 Introduction

OpenID [1] is an open technology standard defining a decentralized authentication protocol, allowing end-users to sign in to multiple websites with the same account. Hence, users maintain their private information in a single point, deciding who is able to obtain such information. Due to its decentralized nature, OpenID does not rely on any central authority validating the trust level of the entities involved in the authentication process. Thus, users can barely know whether a given service is trustworthy enough to share their private information.

A decentralized reputation framework to be integrated with OpenID was defined in [2]. It describes how the OpenID protocol can be enhanced to allow the OpenID provider to collect recommendations about a service, in order to provide useful information about that service to the users beforehand. However, there are several ways of computing the reputation in this environment, becoming a hard task to analyze their feasibility if they are only theoretically described.

Our main contribution in this paper is presenting a simulation environment, developed within NEC Laboratories Europe, able to analyze and validate the feasibility of reputation models integrated with OpenID, such as [2]. This simulator environment, entitled ROMEO (ReputatiOn Model Enhancing OpenID Simulator), allows evaluating, among others, the capability of adversaries to exploit the reputation framework. For instance, analyzing whether a service could unfairly increase its reputation by introducing biased recommendations [3].

S. Mauw and C. Damsgaard Jensen (Eds.): STM 2014, LNCS 8743, pp. 193–197, 2014.
© Springer International Publishing Switzerland 2014

The remainder of the article is organized as follows. Section 2 introduces threats to consider when analyzing this kind of systems. Section 3 describes the internal components defining the architecture of the simulator, whereas Section 4 presents its user interface. Finally, Section 5 presents some concluding remarks.

2 OpenID-Integrated Reputation Frameworks Threats

There are several aspects affecting the behavior of reputation frameworks, which any simulator should consider. Next, we list some relevant assumptions and threats to contemplate when developing an OpenID-based reputation framework.

- Relying parties may try to figure out the recommendation that each OpenID provider has about them.
- Relying parties could offer services with diverse qualities.
- Quality of the services offered by the relying parties could fluctuate.
- Relying parties could fake the list of potential recommenders by including just the recommenders providing better recommendations about them.
- A relying party could decide not to participate in the reputation framework.
- There could be malicious OpenID providers supplying inaccurate recommendations values trying to distort the reputation of a given relying party.
- Users could provide inaccurate or biased recommendations [4].
- The framework cannot assume unlimited resources.
- A malicious entity can present multiple identities, issuing a higher fraction of the recommendations of the system, as a kind of Sybil attack [5].

3 ROMEO Architecture Overview

Internal components of the simulator are shown in Figure 1. The architecture has been designed to allow easy extensibility, in order to validate any reputation computation engine that may be defined in the future targeting OpenID.

- **Reputation Authority.** This component is the entry point for requesting recommendations. Users or other OpenID providers can send queries to this module to obtain the recommendations about a relying party.
- **Rule Engine.** The Rule Engine component aims to influence the reputation computation process according to the defined rules. An example of rule is: *if the system is overloaded, only the last 25 recommendations are considered.*
- **Preferences Engine.** To provide customized recommendations, this component processes the preferences of the users to influence the weights of the recommendations of other users when computing the reputation score [2].
- **Reputation Manager.** The Reputation Manager coordinates the reputation computation. It sends the gathered recommendations to a specific Computation Engine to obtain an aggregated reputation value.
- **Reputation Store.** This component is in charge of maintaining recommendations gathered in the past. Additionally, this module may act as a cache by storing already aggregated reputation values during a certain period of time.

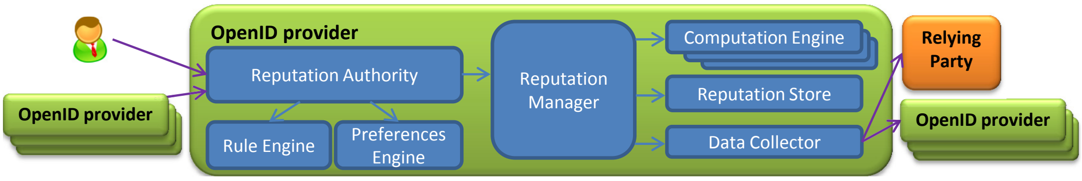


Fig. 1. General ROMEO architecture overview

- **Data Collector.** This component retrieves the list of potential recommenders from a given relying party and then asks each of these recommenders (i.e. other OpenID providers) for recommendations about the relying party.
- **Computation Engines.** These components are in charge of aggregating reputation values from the recommendations. As they follow a common interface, the Reputation Manager could decide which one to use on-the-fly [6].

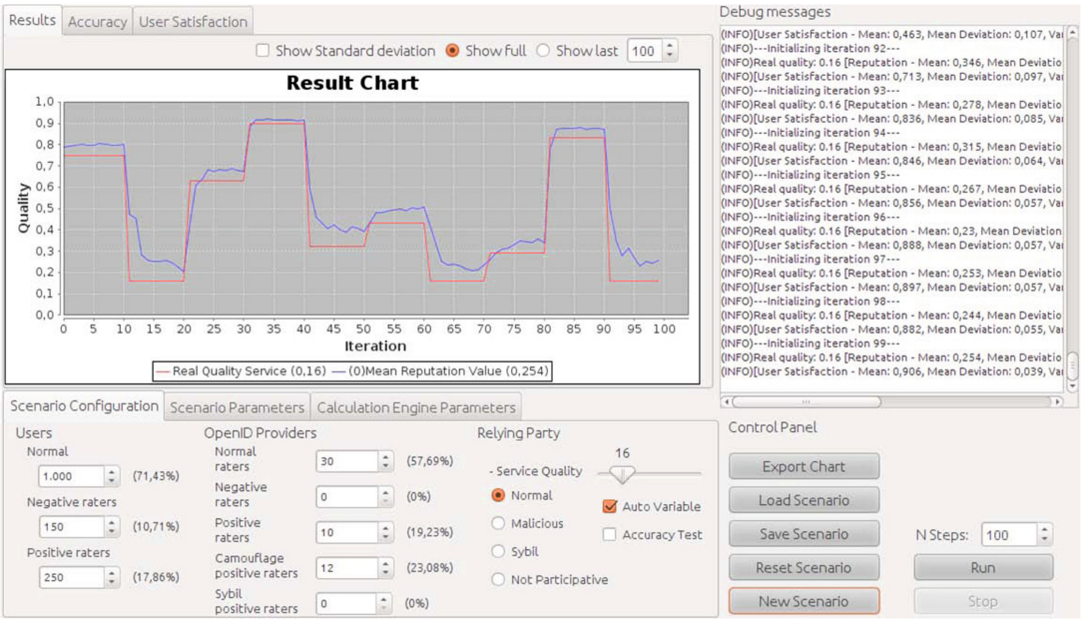
4 Reputation Model Enhancing OpenID Simulator

ROMEO is a simulator created at NEC Laboratories Europe and aimed to evaluate reputation frameworks integrated with OpenID, such as [2]. It allows, among others, analyzing the capability of malicious users or entities (or groups of them) to exploit reputation system vulnerabilities, such as those presented in Section 2. Malicious users or entities mainly aim to distort the reputation of a given relying party by supplying biased recommendations. ROMEO evaluates how certain reputation computation engines behave against different scenarios and threats.

4.1 Scenario Elements

As shown in Figure 2, ROMEO presents a graphical user interface where different reputation-based scenarios and their properties could be defined. A scenario is composed of a set of simulated users interacting with a relying party, by using simulated OpenID providers. The scenario properties define the behavior of the elements in the framework, in order to model the aspects described in Section 2 defining, for instance, whether (and how) the relying party will vary its QoS. It also allows configuring the different reputation computation engines. As part of the scenario properties, we have included the following.

- Type of Users:
  - **Normal:** These users provide appropriate recommendations according to the relying party quality of service.
  - **Negative/Positive Raters:** These users always provides bad (negative raters) or good (positive raters) recommendations when giving feedbacks, regardless of the quality of the received service.
- Type of OpenID Providers:
  - **Normal:** These providers properly follow the reputation framework guides.



Debug messages

```
(INFO)[User Satisfaction - Mean: 0,463, Mean Deviation: 0,107, Var
(INFO)---Initializing iteration 92---
(INFO)Real quality: 0.16 [Reputation - Mean: 0,346, Mean Deviatio
(INFO)[User Satisfaction - Mean: 0,713, Mean Deviation: 0,097, Var
(INFO)---Initializing iteration 93---
(INFO)Real quality: 0.16 [Reputation - Mean: 0,278, Mean Deviatio
(INFO)[User Satisfaction - Mean: 0,836, Mean Deviation: 0,085, Var
(INFO)---Initializing iteration 94---
(INFO)Real quality: 0.16 [Reputation - Mean: 0,315, Mean Deviatio
(INFO)[User Satisfaction - Mean: 0,846, Mean Deviation: 0,064, Var
(INFO)---Initializing iteration 95---
(INFO)Real quality: 0.16 [Reputation - Mean: 0,267, Mean Deviatio
(INFO)[User Satisfaction - Mean: 0,856, Mean Deviation: 0,057, Var
(INFO)---Initializing iteration 96---
(INFO)Real quality: 0.16 [Reputation - Mean: 0,23, Mean Deviation
(INFO)[User Satisfaction - Mean: 0,888, Mean Deviation: 0,057, Var
(INFO)---Initializing iteration 97---
(INFO)Real quality: 0.16 [Reputation - Mean: 0,253, Mean Deviatio
(INFO)[User Satisfaction - Mean: 0,897, Mean Deviation: 0,057, Var
(INFO)---Initializing iteration 98---
(INFO)Real quality: 0.16 [Reputation - Mean: 0,244, Mean Deviatio
(INFO)[User Satisfaction - Mean: 0,882, Mean Deviation: 0,055, Var
(INFO)---Initializing iteration 99---
(INFO)Real quality: 0.16 [Reputation - Mean: 0,254, Mean Deviatio
(INFO)[User Satisfaction - Mean: 0,906, Mean Deviation: 0,039, Var
```

Users
Normal: 1.000 (71,43%)
Negative raters: 150 (10,71%)
Positive raters: 250 (17,86%)

OpenID Providers
Normal raters: 30 (57,69%)
Negative raters: 0 (0%)
Positive raters: 10 (19,23%)
Camouflage positive raters: 12 (23,08%)
Sybil positive raters: 0 (0%)

Relying Party
- Service Quality: 16
● Normal
○ Malicious
○ Sybil
○ Not Participative
☑ Auto Variable
□ Accuracy Test

N Steps: 100

Fig. 2. ROMEO simulator graphical interface screenshot

  - **Negative/Positive Raters:** These providers always give bad/good recommendations about the relying party, regardless of its real behavior.
  - **Camouflaged Positive/Negative Raters:** Extending the previous one, these providers give good/bad recommendation, but only a $p\%$ of the times. The reminder $(100 - p)\%$, they act as normal raters.
  - **Sybil Positive/Negative Raters:** These providers act as positive/negative raters, although after a while, they replace their identity with a new one.
- Type of Relying Party:
  - **Normal:** The relying party properly follows the reputation protocol.
  - **Malicious:** The relying party includes in the recommenders list only the ones with better recommendations about itself.
  - **Sybil:** The relying party is disconnected and replaced with a new identity from time to time, reinitializing its associated reputation.
  - **Not Participative:** The relying party evades the recommender list.

Once defined the scenario, the simulation consists of executing a number of iterations. In each iteration, some of the simulated users ask their OpenID provider for the reputation of the relying party. Hence, the OpenID provider collects and aggregates recommendations using the elements described in Section 3. Depending on the reputation of the relying party, the users interact (or not) with the relying party. Finally, the users provide recommendations about the received service to their OpenID provider, which will be used for subsequent aggregation.

4.2 Visualization of Results

After running a simulation, ROMEO shows three charts, representing three different ways of analyzing the results. These charts are described in the following.

- **Results Chart.** The Results chart (Figure 2) compares the real relying party QoS with the reputation computed by the OpenID providers. This chart aims to evaluate the behavior of the reputation model against a specific scenario.
- **Accuracy Chart.** Taking into account that the users interact with the relying party with a probability p, being p the reputation given by its OpenID provider, this chart determines how many users interact with a given relying party. This chart is useful to compare different reputation computation engines regarding their accuracy when calculating reputation scores.
- **User Satisfaction Chart.** It indicates how satisfied the users are with the reputation values they receive. Users' satisfaction is higher if the reputation values fit the quality they receive from the service, according to their preferences. This chart aims to compare reputation models regarding the adaptation to users preferences.

5 Conclusions

The OpenID standard defines a decentralized authentication initiative. As such, OpenID does not rely on any central authority, which makes the trust of the involved entities hard to validate. Some research has been conducted to mitigate this problem. However, the proposed solutions need a deep analysis and validation. In this paper we have described a simulation environment able to evaluate the feasibility of reputation frameworks in this context, and analyzed their behavior within different scenarios. The simulator allows analyzing reputation models against reputation-related threats, involving malicious users or entities.

References

1. Recordon, D., Reed, D.: OpenID 2.0: a platform for user-centric identity management. In: Proceedings of the Second ACM Workshop on Digital Identity Management, DIM 2006, pp. 11–16 (2006)
2. Dólera Tormo, G., Gómez Mármol, F., Martínez Pérez, G.: Towards the Integration of Reputation Management in OpenID. Computer Standards & Interfaces 36(3), 438–453 (2014)
3. Gómez Mármol, F., Martínez Pérez, G.: Security Threats Scenarios in Trust and Reputation Models for Distributed Systems. Elsevier Computers & Security 28(7), 545–556 (2009)
4. Borg, A., Boldt, M., Carlsson, B.: Simulating malicious users in a software reputation system. In: Park, J.J., Lopez, J., Yeo, S.-S., Shon, T., Taniar, D. (eds.) STA 2011. CCIS, vol. 186, pp. 147–156. Springer, Heidelberg (2011)
5. Douceur, J.R.: The sybil attack. In: Druschel, P., Kaashoek, M.F., Rowstron, A. (eds.) IPTPS 2002. LNCS, vol. 2429, pp. 251–260. Springer, Heidelberg (2002)
6. Dólera Tormo, G., Gómez Mármol, F., Martínez Pérez, G.: Dynamic and flexible selection of a reputation mechanism for heterogeneous environments. In: Future Generation Computer Systems (June 2014)

5

Dynamic and Flexible Selection of a Reputation Mechanism for Heterogeneous Environments

Title:	Dynamic and Flexible Selection of a Reputation Mechanism for Heterogeneous Environments
Authors:	Ginés Dólera Tormo, Félix Gómez Mármol, Gregorio Martínez Pérez
Type:	Journal
Journal:	Future Generation on Computer Systems, Special Issue on Trustworthy Data Fusion and Mining in Internet of Things
Publisher:	Elsevier
Year:	2014
DOI:	`http://dx.doi.org/10.1016/j.future.2014.06.006`
State:	In Press, available online

Table 5: Dynamic and Flexible Selection of a Reputation Mechanism for Heterogeneous Environments

Future Generation Computer Systems ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Dynamic and flexible selection of a reputation mechanism for heterogeneous environments

Ginés Dólera Tormo[b,*,1], Félix Gómez Mármol[a], Gregorio Martínez Pérez[b]

[a] *NEC Europe Ltd., Kurfürsten-Anlage 36, 69115 Heidelberg, Germany*

[b] *Department of Information and Communications Engineering, University of Murcia, Murcia, 30100, Spain*

HIGHLIGHTS

- Design a flexible mechanism to select the most suitable trust and reputation model to apply on-the-fly.
- Selection of the reputation model based on current system conditions and expected performance measurements.
- The approach guarantees a smooth and automatic transition between the reputation computation engines.
- Experiments conducted to validate the feasibility of the approach.
- Focus on heterogeneous Internet of Things environments.

ARTICLE INFO

Article history:
Received 4 April 2014
Received in revised form
3 June 2014
Accepted 14 June 2014
Available online xxxx

Keywords:
Trust and reputation management
Internet of things
Interchangeable computation engine

ABSTRACT

Current trust and reputation management approaches usually offer rigid and inflexible mechanisms to compute reputation scores, which hinder their dynamic adaptation to the current circumstances in the system where they are deployed. At most, they provide certain parameters which are configurable or tunable. Yet, this is not enough for such heterogeneous and dynamic environments as the ones introduced by Internet of Things (IoT). In this paper we propose a rupture with this old philosophy and have therefore designed and prototyped a flexible mechanism to select the most suitable trust and reputation model to apply on-the-fly, amongst a pool of predefined ones, considering both the current system conditions and the selected performance measurements, which, to the best of our knowledge, is missing nowadays. Additionally, this mechanism guarantees a smooth transition between different computation engines avoiding abrupt changes in the computed reputation scores. Conducted experiments prove that our solution is able to identify and start up the most suitable trust and reputation model depending on the current system conditions (number of users, allocated resources, etc.) and expected performance measurements (accuracy, scalability, robustness, etc.).

© 2014 Elsevier B.V. All rights reserved.

1. Introduction

Trust and reputation management systems are widely spread and used today in the Internet. We find them in a myriad of service provisioning scenarios, ranging from pure e-Commerce ones to blogs, social networks, video streaming services, etc. [1–3]. Furthermore, an extensive amount of research work has been performed as well in applying trust and reputation management techniques to P2P networks [4], wireless sensor networks [5], vehicular ad hoc networks [6], Cloud Computing [7], Identity Management systems [8], collaborative intrusion detection networks [9,10], etc.

Nevertheless, though this large variety of systems and scenarios constitute a proof of the applicability and feasibility of trust and reputation management solutions, they also lead to a so far neglected problem raised by widely dynamic environments as the ones introduced by Internet of Things (IoT). In IoT environments, many heterogeneous devices (i.e. widely having dissimilar elements, features or behaviors) define dynamic, complex and distributed frameworks [11–13]. Obviously, each system/scenario has different and specific requirements and particularities in terms of infrastructure design, participating entities, communication capabilities, exchanged data, etc. Even

[*] Corresponding author. Tel.: +34 868 887646; fax: +34 868 884151.
E-mail addresses: ginesdt@um.es (G. Dólera Tormo), felix.gomez-marmol@neclab.eu (F. Gómez Mármol), gregorio@um.es (G. Martínez Pérez).

[1] The work presented in this paper was performed while working at NEC Laboratories Europe.

http://dx.doi.org/10.1016/j.future.2014.06.006
0167-739X/© 2014 Elsevier B.V. All rights reserved.

more, the scenarios might be dynamically and continuously changing (their topology, their committed resources, etc.).

Current trust and reputation models usually provide certain configuration parameters aimed to tune the behavior of the deployed mechanism. However, this settings feature is quite often not able to offer the high dynamicity required to adapt the model to different circumstances that may happen in IoT environments. As shown in [14] each trust and reputation model has its advantages and shortcomings, and most of the times their drawbacks are related to the current conditions of the system [15] (number of participants, number of feedbacks, feedbacks storage capabilities, computational capabilities, etc.). Moreover, the expected performance measurements (accuracy, robustness, resilience against attacks, etc.) also affect on the selection of the most appropriate reputation models, and they usually depend on the requirements of the scenario.

Therefore, a flexible mechanism to select the most suitable trust and reputation model to apply on-the-fly considering both the current system conditions and the expected performance measurements, to the best of our knowledge, is missing today. Notwithstanding such fact, we should not ignore the interoperability between models, since every model might not be applicable to all scenarios. However, reputation scores are usually calculated through a module within the model, known in the literature as *reputation computation engine*, which implements different algorithms to aggregate recommendations depending on the scenario.

In order to tackle the aforementioned problem, we have designed and prototyped a system [16] which is able to dynamically select the most appropriate reputation computation engine according to the current conditions of the system (in terms of number of users, number of feedbacks, available bandwidth, available storage capacity, etc.), as well as to the desired performance measurements (i.e., accuracy, user satisfaction, adaptability, resilience to certain attacks, etc.), aimed to deal with the dynamicity introduced by IoT. The suitability of each reputation computation engine is computed based on predefined inference rules, which relate system conditions to performance measurements. These rules are defined using fuzzy sets [17], in order to improve the flexibility in the rule definition process, usually performed by system administrators. Additionally, our approach guarantees a smooth transition between different computation engines, avoiding abrupt changes in the computed reputation scores.

Applying trust and reputation to IoT is no longer a matter of designing and developing tunable trust and reputation models, but to provide a pool of them, analyzing their intrinsic characteristics in order to determine under which conditions they provide the best outcomes for each of the desired performance measurements. In this way, knowing which are the performance metrics required by each scenario, and monitoring the system conditions, the system is able to dynamically select the most suitable model according to the parameters of such a scenario. The models are automatically swapped in order to have the most suitable reputation model working at each moment.

Furthermore, how to measure the suitability could depend on the scenario, since the performance metrics to be optimized might vary from one environment to another. For instance, a target of a reputation model within a IoT-based sensors environment could be to minimize the consumption of required resources [18], whereas for a IoT cloud ecosystem the target could be to improve the accuracy of the applied trust and reputation mechanisms no matter how much computation is required [19].

The remainder of this article is structured as follows. Section 2 presents a description of the problem being addressed in this work. Then, in Section 3 we introduce our solution for dynamically selecting the most appropriate reputation model on-the-fly, while in Section 4 a mechanism to avoid abrupt changes when reputation computation engines are swapped is described. Section 5 shows some conducted experiments in order to validate the feasibility of the proposed system. Later, Section 6 provides the main references and related works, while in Section 7 the main conclusions and lines of future work are outlined.

2. Problem statement

IoT is predicted to revolutionize the way organizations implement their information systems and applications [20]. Expecting tons of devices seamlessly integrated into information networks, IoT enables unlimited scalability and greater flexibility all at a contained cost. It introduces several new business models based on unlimited application scenarios and new smart services. On the other hand, IoT raises new challenges, and companies and organizations supporting this concept have to face a number of burdens and barriers to its deployment. Trust, at the core of these concerns, is identified as a critical component to allow the IoT to reach its greatest potential [12,21].

In such a dynamic and distributed environment, static agreements are generally hard to apply for managing trust relationships between different entities, such as those based on Service-Level Agreements (SLA), where rigid (and usually complex) contracts are applied. Reputation management systems have emerged in the last years as a solution for this kind of scenarios, where the trust of a given entity is dynamically acquired from analyzing its past and recent behavior.

Reputation management systems propose mechanisms to allow an entity to somehow determine if another entity can be taken as reliable or not, in order to get some services or exchange some information between them. Firstly, these systems try to collect as much information as possible about the behavior of a given entity, which usually comes from recommendations based on past experiences. Secondly, all gathered recommendations are aggregated in order to calculate a reputation score for such an entity [14].

The computed score will be used to decide the level of trustworthiness that a particular entity has. If such entity has enough reputation, the communication process is therefore triggered between the entities. Finally, the service consumer provides the satisfaction with regards to the received service which, in turn, will be used as a recommendation for calculating future reputation scores.

For instance, we can think of an IoT scenario in the eHealth context, where information about patients is collected through heterogeneous sensors, in order to track their status. These sensors might be deployed through complex network structures, where the trust information could be hardly managed in a single point. As an alternative, reputation management systems could collect recommendations from different sources, to decide whether a given entity (e.g. a sensor, a set of them, an intermediary node, or even a network path [22]) is trustworthy enough to be in charge of such a sensitive data.

Reputation management systems have been proposed in different contexts and they have been applied to many different scenarios [1]. Since the behavior of the reputation system mainly depends on the requirements of the scenario where such system is deployed, a lot of mechanisms to accomplish the aforementioned steps have been defined. For instance, depending on the computation constraints of the devices which aggregate and compute the reputation values, a different reputation computation engine might be used, whereas a different way of collecting recommendations may be used depending on their network capabilities.

Due to the multitude of different reputation management systems nowadays, it is difficult for an administrator to choose which one fits better within a concrete IoT scenario. Furthermore, once a reputation management system has been adapted and applied to the scenario, replacing it by another one constitutes a cumbersome task. At most, these reputation management models provide a number of parameters which are configurable or tunable, but they rarely allow substantial changes on-the-fly. In case the administrators want to change the reputation model currently in play, they would need to configure or even re-adapt the systems they are managing to work with the new reputation management model, since different reputation models would require different inputs.

This lack of adaptation is further emphasized on IoT environments, where the requirements of services, involved entities and number and nature of users is highly changeable and unpredictable [11]. Since there is not a unique reputation model suitable for every condition, the frequent changes in the elements of the scenario in IoT environments may make necessary to be swapping the reputation model for a more appropriate according to the current conditions.

It could also result in having an inconsistent period while the swap is being performed, since the new system would require a start-up time before providing accurate reputation values. This would be unacceptable for some IoT scenarios, as the eHealth example presented above, where accuracy must be maximized while delays and administration costs should remain as low as possible. However, as far as we know, a flexible mechanism to select the most suitable trust and reputation model to apply on-the-fly, taking into account current system conditions and expected performance measurements, is not present today.

3. Dynamic and flexible selection of a reputation engine

A number of trust and reputation models can be found in the literature, proposing different ways of calculating the reputation of a given entity from past interactions. The faithfulness of the selected model depends on several factors, which usually change throughout time, such as the level of participation of the sources which provide recommendations, the network load, the percentage of malicious nodes feeding the system with biased recommendations, etc. [15]. This section will present a solution that is able to dynamically select the most appropriate reputation model, in a flexible way.

3.1. Overview

As previously described, reputation management systems are aimed to predict the behavior of a given entity or service by analyzing its past behavior. To that end, they collect recommendations about such entity or service from different sources which have already had some experience with it. These recommendations are aggregated making use of a specific aggregation algorithm, which may also take into consideration contextual information available at the time of computation. As a result, they obtain the reputation value of the entity or service, which represents its expected behavior. Following the IoT eHealth scenario example, where different sensors collect patients' information, a straightforward reputation model would gauge the reputation value of an specific sensor by calculating the arithmetic mean of all the available recommendations that other sensors have about it.

In order to differentiate the aggregation process from the rest of the processes of the reputation model, such as gathering information, storage management, punishment to malicious entities, etc., the internal module of any reputation model in charge of executing the aggregation algorithm to compute a final reputation score was isolated in [14] and called *Reputation computation engine*. In other words, the reputation computation engine takes recommendations and other contextual information about a given entity as input and outputs the reputation of such entity as a result.

In this way, we can consider that the core behavior of reputation models lies on how they implement their respective reputation computation engines. For example, whether this reputation computation engine deploys mechanisms to avoid malicious users, or if it takes into account network load, or even if it would provide personalized outcomes [23].

With the aim to select which reputation model fits better for a specific IoT environment and conditions, administrators or designers need to analyze the intrinsic properties of each reputation computation engine available. For example, they estimate the number and type of IoT devices that would be deployed and how they will interact with each other. They determine under which conditions these models provide the best outcomes for each of the desired performance measurements, such as minimize sensor energy consumption or network load. Then, they choose a reputation model, and adapt their system to work with it. Furthermore, to analyze the properties of the models, they may want to use data mining techniques, aimed to detect patterns in the behavior of the reputation models, so it provides understandable information for helping in the selection decision [24].

There might be some scenarios where it is hard to analyze the intrinsic characteristics of the available model, or even it could be the case where only one model is applicable. In those cases, the administrator could decide to have a tunable model which might achieve modest results. However, this solution is focused on those scenarios where the administrators have information about the behavior of the different models, yet not able to decide which one fits better in this scenario.

Instead of forcing the administrators to chose one reputation computation engine depending on the initial requirements of the IoT scenario, our solution provides a pool of reputation computation engines controlled by the *Engine Selector*, as shown in Fig. 1. The *Engine Selector* maintains just one of the reputation computation engines as active, which means that it is computing the reputation values when required, while the rest remain on an idle state, hence not consuming computational resources.

In parallel, a number of system conditions is being monitored and analyzed together with the performance metrics required by the scenario. These parameters and the properties of each reputation computation engine are used as the input for the selection mechanism. That selection is performed based on inference rules making use of fuzzy sets, which provide a flexible way to ascertain which reputation computation engine is the best candidate to become active, as described in Section 3.2.

In the IoT eHealth example scenario, the system would require to change the reputation model when the topologies of the networks change. For instance, it could happen that previous sources of recommendations are no longer accessible, or that the active reputation computation engine counts on less available computer or network resources to collect and compute the recommendations. In this case, it could be beneficial to select a reputation computation engine requiring less recommendations and/or resources to properly work, in order to prevent the accuracy of the reputation values to drop.

As soon as the reputation framework asserts that a particular idle reputation computation engine could provide better outcomes than the current active one, then the latter is swapped with the former. Yet, the exchange of reputation computation engines is undertaken gradually, that is, taking inputs from different engines at the same time, enabling a smooth transition as described in Section 4.

The reputation computation engine selection process is triggered eventually to determine if a swapping has to be performed

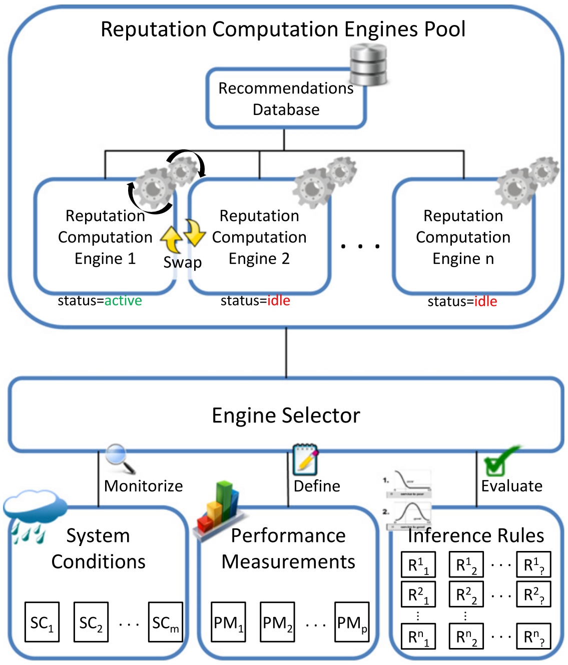

Reputation Computation Engines Pool
Recommendations Database
Reputation Computation Engine 1
Swap
Reputation Computation Engine 2
. . .
Reputation Computation Engine n
status=active
status=idle
status=idle
Engine Selector
Monitorize
Define
Evaluate
System Conditions
SC_1 SC_2 . . . SC_m
Performance Measurements
PM_1 PM_2 . . . PM_p
Inference Rules
R^1_1 R^1_2 . . . $R^1_?$
R^2_1 R^2_2 . . . $R^2_?$
R^n_1 R^n_2 . . . $R^n_?$

Fig. 1. Architecture overview of the dynamic reputation computation engine selector.

(i.e. if there is a better model to become active). This process could be initialized periodically, for instance, after a certain amount of time defined by the administrator. Additionally, that process could be triggered after certain events related to the current system conditions. That is, as the system conditions are being monitored, the system is able to detect whether some conditions have varied in a certain amount, triggering the selection process as there might be a better candidate under those new conditions.

Furthermore, the selection process is also executed during the bootstrapping of the reputation framework, in order to determine which reputation computation engine is active as soon as the system starts working. The input data (e.g. monitored system conditions) used for this selection would be limited as the system has not collected any information yet. However, some default values could be used until information about any system condition is available. Thus, a model requiring less amount of recommendations or users' participation would be likely chosen at the beginning.

It is worth mentioning that recommendations and other gathered information are maintained in a common database, in order to allow all the reputation computation engines to get the information they need to compute reputation values. Nevertheless, data specific to each reputation computation engine is maintained individually, like for instance, in case a reputation computation engine needs to store additional information about the users (e.g. users preferences, weights to measure their goodness, etc.) to enhance the accuracy of their reputation values.

3.2. Selection of the most suitable candidate

In order to judge which reputation computation engine is the best candidate to aggregate the recommendations at each moment, we define Algorithm 1, which receives the m current system conditions $(SC_1, SC_2, \ldots, SC_m)$ and the p desired performance measurements $(PM_1, PM_2, \ldots, PM_p)$ as inputs and deterministically provides a set of values representing the suitability of the n available reputation computation engines $(ST_1, ST_2, \ldots, ST_n)$.

```
Input: $(SC_1, SC_2, ..., SC_m), (PM_1, PM_2, ..., PM_p)$
for $k \leftarrow 1$ to $n$ do
    $(PM^k_1, PM^k_2, ..., PM^k_p) \leftarrow$
        $fuzzy\_inference(SC_1, SC_2, ..., SC_m)$;
    $ST_k \leftarrow vector\_distance((PM^k_1, PM^k_2, ..., PM^k_p),$
        $(PM_1, PM_2, ..., PM_p))$;
end
Output: $(ST_1, ST_2, ..., ST_n)$
```

Algorithm 1: Algorithm to compute the suitability of the reputation computation engines

The *fuzzy_inference*() function used within the algorithm obtains the assessed performance measurements $(PM^k_1, PM^k_2, \ldots, PM^k_p)$ the reputation computation engine k (RCE_k) is supposed to offer with the current system conditions according to a number of predefined inference rules. The algorithm also makes use of the *vector_distance*() function to determine whether the reputation computation engine would provide the pursued performance metrics. In fact, this function computes the Euclidean distance between the expected performance measurements and the assessed ones. In that way, the suitability of the reputation computation engines can be measured and hence compared between each other.

As a verbose summary, the defined algorithm takes the system conditions which are being monitored and it evaluates the inference rules for each computation engine. By doing so, it determines the distance between the performance measurements which would be obtained by each computation engine (according to the inference rules), and the desired performance measurements. The less distance (i.e. expected performance measurements are closer to the desired ones), the more suitable to become active the candidate is.

As a straightforward example, let us suppose we have two reputation computation engines, RCE_1 and RCE_2, and we have defined the following simplified inference rules.

1. When the amount of IoT sensors is low
   (a) RCE_1 presents low energy consumption and medium accuracy.
   (b) RCE_2 presents medium energy consumption and medium accuracy.
2. When the amount of IoT sensors is high
   (a) RCE_1 presents medium energy consumption and medium accuracy.
   (b) RCE_2 has medium energy consumption and high accuracy.

In this example, the amount of IoT sensors is a system condition, whereas both the level of energy consumption and the accuracy are performance measurements. Let us also suppose that the desired outcome is having low consumption and high accuracy. On one hand, if the current number of IoT sensors is low, the reputation computation engine with higher suitability would be RCE_1, because the expected outcome would be closer to the desired one. On the other hand, if the amount of IoT sensors is high, RCE_2 would have higher suitability.

In turn, the inference rules, which are the base of the inference process [25], relate an antecedent to a consequent, both defined using fuzzy sets [17,26]. The antecedent is composed of a fuzzy set for each system condition, modeling a membership function to determine the level of applicability of each system condition when the rule is evaluated. The consequent is composed of a fuzzy set for each of the performance measurements, indicating what the reputation computation engine would offer under the system conditions that the antecedent indicates.

Using fuzzy sets to define the rules instead of classic sets (i.e. those where an element either belongs or not to the set)

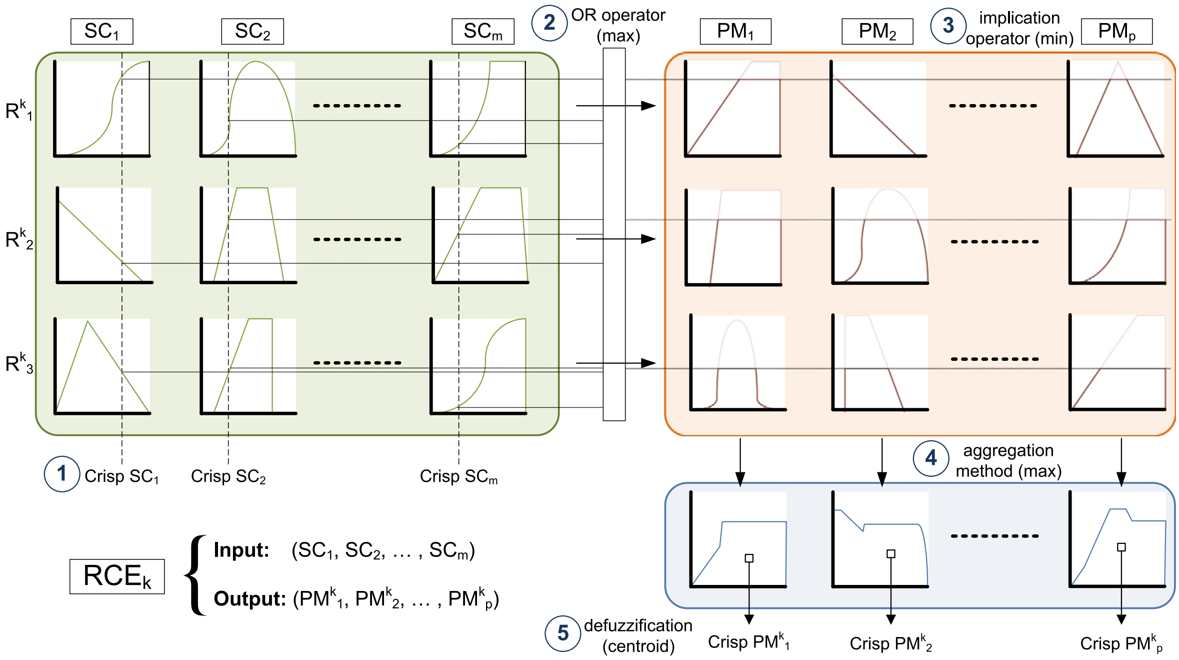

SC₁
SC₂
SCm
2 OR operator (max)
PM₁
PM₂
3 implication operator (min)
PMp
Rk1
Rk2
Rk3
1 Crisp SC₁
Crisp SC₂
Crisp SCm
4 aggregation method (max)
RCEk
Input: (SC₁, SC₂, … , SCm)
Output: (PMk1, PMk2, … , PMkp)
5 defuzzification (centroid)
Crisp PMk1
Crisp PMk2
Crisp PMkp

Fig. 2. Fuzzy inference process based on rules which relate system conditions to performance measurements.

is mainly aimed to improve the flexibility in the rule definition process, since they allow defining a gradual assessment of the membership of elements in a set. However, with the aim of easing the interpretability of the inference rules, the fuzzy sets are modeled as elements from a set of linguistic labels [27,28]. They provide an easy and intuitive comprehension to administrators with no skills on data mining techniques. For instance, the values of each of the system conditions could be tagged as "very low", "low", "medium", "high" and "very high". In this way, the rules could be defined in a human comprehensible way, making such definition easier for administrators. For instance, to compose the inference rules, an administrator could specify things like a reputation computation engine has "high" accuracy and "very high" resilience to attacks when the number of users is "medium".

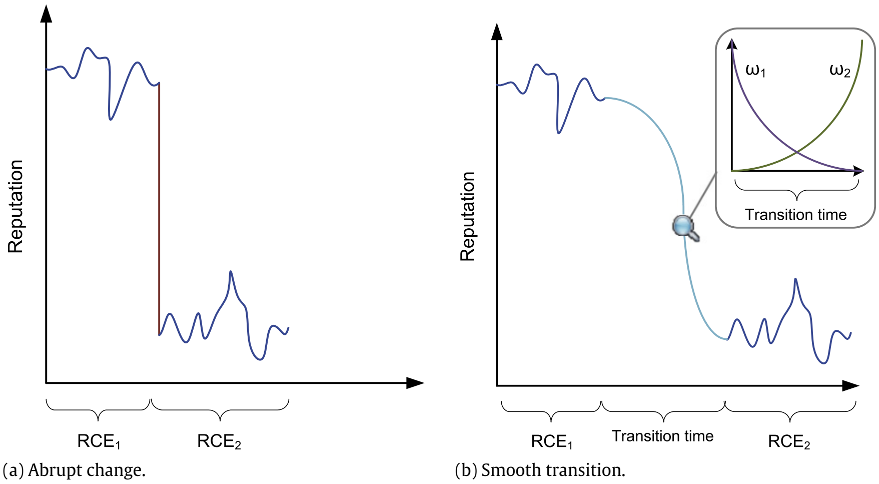
Moreover, if some of the parameters of the antecedent (system conditions) or the consequent (performance measurements) are not known for a given rule, they would be tagged as "unknown". In that case, the unknown parameters would be represented by a default fuzzy set defining a default membership function aimed to model the average behavior of that parameter. In other words, the "unknown" would be ideally defined in such a way that it neither punishes nor benefits that parameter over the rest.

Defining the inference rules in this way allows extensibility, in such a way that additional system conditions or performance measurements could be easily added on demand. For instance, it would be the case that after having the system running for a while, an administrator wanted to include a system condition or performance measurement in order to influence the selection process.

To handle that, the administrator would just need to extend the system conditions vector $(SC_1, SC_2, \ldots, SC_m)$ or the performance measurements vector $(PM_1, PM_2, \ldots, PM_p)$ for each of the defined rules in order to incorporate the new system condition $(SC_1, SC_2, \ldots, SC_m, SC_{m+1})$ or performance measurement $(PM_1, PM_2, \ldots, PM_p, PM_{p+1})$. Note that, for those rules where the new parameter is not known, the "unknown" tag could be used.

Fig. 2 shows an example of three inference rules $(R^k{}_1, R^k{}_2$ and $R^k{}_3)$ defined for the reputation computation engine k (RCE_k), following the aforementioned mechanism (for simplicity, the linguistic labels have been omitted). The fuzzy inference process, also shown in Fig. 2, starts with the crisp numerical values of the current system conditions. The following steps are performed.

1. **Fuzzify inputs**: From the system conditions' crisp values, we compute the degree to which each element of the antecedent is satisfied for each rule. This degree is calculated from the fuzzy sets membership function which has been defined for each system condition in that rule. In other words, each rule applies the membership function which was defined for each system condition using the current system conditions' crisp values as input. Consequently, a set of values (one value per system condition) is obtained for each rule, identifying how much this rule should be applied.
2. **Apply fuzzy operator**: The objective of this step is to obtain, for each rule $R^k{}_i$, a number representing the matching degree of the whole antecedent of such rule with the crisp values used as input. As we mentioned, the result of the previous step is a set of values for each rule. Thus, the idea is to aggregate somehow those values to get a single value for each rule. To that end, a fuzzy operator is applied. In our case we perform the OR operation to obtain the greatest value, although other alternatives could be applied, such as getting the smallest, compute the average value, etc. Furthermore, after applying the operator, a weight might be applied to the resulting value to give some of the rules higher or lower relevance.
3. **Apply implication operator**: The consequent of each rule was also defined as a set of fuzzy sets represented by a number of membership functions, respectively, modeling the expected performance measurements. In this step, and for each rule, using the value obtained from the previous step, all the fuzzy sets of the consequent are reshaped, obtaining different fuzzy sets. In our case, the *minimum* (AND) method is applied, which truncates the fuzzy sets defined for the assessed performance measurements. This step is performed in order to represent how the dissimilarity of the antecedent from the current system conditions affects the defined consequent. Note that, if the current system conditions match the antecedent perfectly, the output of this step would be the consequent itself.



(a) Abrupt change.

(b) Smooth transition.

Fig. 3. Transition between reputation computation engines.

4. **Apply aggregation method**: The idea of this step is to aggregate the reshaped consequent of the rules in order to provide a single vector of fuzzy sets as output. Thus, for each performance measurement, all the fuzzy sets resulting from the previous step are aggregated to form a new fuzzy set. That is, the membership functions that define the fuzzy sets are combined to form a new membership function. Our solution makes use of the *maximum* method to perform such aggregation.
5. **Defuzzify**: Finally, the defuzzification process outputs a crisp value from a fuzzy set, which is indeed the desired outcome. Again, there are different mechanisms to make this process, such as getting the largest of maximum, or the average of the maximum value. In this approach, the centroid is computed, which returns the center of gravity of the area under the curve defined by the membership function of the corresponding fuzzy set. Finally, a set of values predicting the expected performance measurement given the current system conditions is obtained for each reputation computation engine.

Once the *Engine Selector* executes the algorithm, it obtains the suitability of each of the reputation computation engines, which can be directly compared between each other in order to decide which is the best candidate to become active. Usually, the reputation computation engine with highest suitability is selected to become active, since it has been predicted to provide better outcomes according to the defined rules. Furthermore, it could be the case that the selected engine is the one already active, meaning that no transition is needed.

Additionally, under certain circumstances, the administrators might prefer to configure the selection process in such a way that the reputation computation engine to become active is not always the most suitable, but to perform a probabilistic selection based on those suitability values. For instance, if the administrators do not have enough information to properly define the inference rules, they would prefer at the bootstrapping phase to apply a probabilistic selection. Thus, reputation computation engines which (according to the predefined rules) are not the best candidates under certain system conditions could become active, even though candidates really inadequate would have very low probabilities to become active. In this way, the administrators could analyze the behavior of the different reputation computation engines within the concrete IoT scenario, in order to adapt or specify new inference rules accordingly.

To achieve the probabilistic selection, firstly a normalization of the suitability values has to be carried out, in such a way that each suitability value is between 0 and 1 ($\forall k \in [1, n], ST_k \in [0, 1]$), and the sum of all the suitability values is equal to 1 ($\sum_{k=1}^{n} ST_k = 1$). Thus, the normalized values correspond to the probability of selecting each one of the reputation computation engines.

4. Smooth transition between reputation engines

Once the most suitable reputation computation engine has been chosen, it becomes active and starts computing reputation values. Nevertheless, it might require certain time before outputting reliable reputation scores due to the bootstrapping period, as it usually requires to initialize and stabilize its parameters. For instance, in IoT scenarios a reputation computation engine may assign weights to the recommenders (e.g. sensors, nodes, users, etc.) in order to determine their goodness [29]. In this case, it would firstly require receiving some recommendations to adjust such weights.

On the other hand, having all the reputation computation engines activated (i.e. aggregating and computing reputation) and just selecting the most appropriate would make the reputation system too heavy and even infeasible in many environments. A representative IoT example where this is not practical would be a sensor scenario, where minimizing the energy consumption, and therefore the computational load, is usually a requirement.

Hence, we want to avoid a potential abrupt change in the computed reputation score when shifting the reputation computation engine in use (Fig. 3a). In other words, we need to provide a smooth transition between the preceding reputation computation engine and the selected one (Fig. 3b).

In order to achieve such smooth transition, once the computation engine exchange has been triggered, a so called “transition time” is considered, where both computation engines co-exist and are considered as active. Let reputation computation engine RCE_i, be the current engine and reputation computation engine RCE_k, the selected engine to replace RCE_i, whereas $Rep_{RCE_i,t}$ and $Rep_{RCE_k,t}$ are the reputation values provided at time t by RCE_i and RCE_k, respectively. Then, during this transition period, the global reputation scores (Rep_t) will be computed taking into account the outputs from both engines but assigning a dynamic weight to each of them (ω_i and ω_k, respectively).

In general, the global reputation score during the transition period is computed as a function of the reputation values provided by each reputation computation engine and their weights at each moment, as shown in the following equation.

$$Rep_t = \omega_{i,t} \cdot Rep_{RCE_i,t} + \omega_{k,t} \cdot Rep_{RCE_k,t}. \quad (1)$$

The weights determine how much the output of a specific reputation computation engine influences the global output at a certain moment. Furthermore, such weights ω_i and ω_k are defined in a fashion that, at the beginning of the transition period, $\omega_i = 1$ and $\omega_k = 0$. Then, ω_i will steadily decrease (until reaching $\omega_i = 0$) in the same proportion as ω_k (until reaching $\omega_k = 1$) will increase, as shown in Fig. 3b, always fulfilling $\omega_i + \omega_k = 1$.

Eq. (1) is a simple but representative example, although more complex formulas could be used instead, even taking more parameters into account. In the end, the idea behind the presented equation is that, at the beginning of the transition period, the outputs of the current reputation engine (RCE_i) are uniquely considered, but little by little its influence is decreasing while giving way to the recently selected computation engine (RCE_k), as its outputs are more and more taken into account. Besides, the duration of transition time could also vary dynamically as well as it could depend on some system conditions.

5. Experiments and results

This section describes the experiments conducted aiming to validate the feasibility of the proposed solution. The main point to analyze is whether this solution could produce more accurate reputation values in the long term than those computed by traditional reputation models, where only one reputation computation engine is working throughout the lifetime of the system.

5.1. Reputation computation engines

Different ways to compute reputation values by aggregating collected recommendations were analyzed in [15]. That work also evaluates and compares different reputation computation engines and concludes that there is not an ideal computation engine which produces good outcomes for every condition. Due to the fact that the computation engines evaluated have been proved to provide different outcomes under different system conditions and/or expected performance measurements, we use some of the models presented in [15], in order to undertake our experiments. Note that in these models the term "user" refers to an end-user, that is, a person who uses a service. In IoT, an user could be a person, a device, or any "thing" which could use a service. A summary of these selected models is presented next.

5.1.1. Weighted Average

In order to prevent malicious users aiming to increase or decrease the reputation of a given entity by providing biased recommendations about it, this computation engine assigns a weight to each user, representing his or her goodness. In this sense, the higher is the weight associated to a user the more honest he or she is supposed to be when providing recommendations. The goodness is indeed calculated from the deviation of his or her recommendations when compared to the recommendations provided by the rest of the users. For example, the goodness of a user decreases if she tends to provide good recommendation about some services which are rated as bad by most of the users.

To compute the reputation value, instead of just computing an arithmetic mean taking all the available recommendations with the same relevance, the reputation value is calculated as a weighted average according to those weights. In this sense, the recommendations given by dishonest users (those which provide biased recommendations) are less taken into account, therefore preventing this kind of users.

This model does not require a lot of recommendations to provide accurate reputation values, and does not need many computational resources to work. However, in IoT scenarios where even honest users have a lot of divergence in their opinions (i.e. provide contradictory recommendations about the same services), the accuracy of the reputation framework decreases.

5.1.2. Preferences weighted average

This computation engine tries to provide customized reputation values to the users. Besides the weights indicating the goodness of each user, it computes the similarity between each pair of users. The similarity value is based on the users' preferences, which express the assessment of each user with regards to the properties describing the service. For example, for an IoT network management service, the users could establish their predilection about parameters like usability, performance, average delay, price, etc.

Since there is an additional factor to be taken into account, the Preferences Weighted Average computation engine requires more computational resources than the previous one to calculate the reputation values. Moreover, to optimize its behavior, this computation engine would require a higher amount of users participating in the recommendation system, since having enough recommendations is required for each set of preferences. That is, the users would not receive accurate reputation values if there are not enough users sharing their same preferences and hence not enough recommendations to compute. On the other hand, if the system gets to have enough users' recommendations, this computation engine provides more accurate results than the Weighted Average.

5.1.3. Users Weighted Average

Even users having defined the same preferences might have different opinions with regard to some of the received services. To handle that aspect, this computation engine calculates a weight for each pair of users to represent the similarity between them. These weights are updated according to recommendations provided by those users, in such a way that users providing similar recommendations would have a higher weight. In this sense, when computing the reputation value for a given user, this engine gives higher relevance to the recommendations provided by users with higher similarity to the given one.

The Users Weighted Average computation engine produces highly accurate results in the reputation values. Nevertheless, it requires higher computational resources, and quite a lot of users, which in addition must be very active (i.e. users frequently providing recommendations) to achieve its optimal behavior.

5.2. Experiments settings

In order to evaluate the proposed solution, we have made use of the simulation tool described in [30]. This tool is able to define a reputation-based scenario where users interact with a service according to its reputation, which is computed by a reputation service using a concrete reputation model. Within IoT, a reputation service refers to a service aimed to provide reputation values from collected recommendations, and it could be deployed by a trusted party, or by any device or set of them in a distributed way [31]. This tool allows specifying the number of users requesting the service, as well as their goodness, i.e., whether their recommendations will be accordant to the service received (normal users) or biased either positively or negatively (malicious users). Besides, the tool allows specifying the quality of the service offered, and other scenario parameters, such as whether the quality of the service would vary throughout the time, the percentage of user participation which makes the system have more or less recommendations to compute the reputation, etc. The relevant parameters configured for this testbed are summarized in Table 1.

Once the scenario has been defined, the tool simulates a number of sequential phases, where (i) each user asks the reputation service for computing the reputation value, (ii) the reputation service collects users' recommendations, (iii) the reputation service aggregates the recommendations using one of the reputation computation engines and provides the reputation

Table 1
Experiment tool settings.

Parameter	Value	Explanation
Number of users (recommenders)	10,000	Enough number of users to simulate a complex IoT scenario
Percentage of malicious users	30%	Introducing a noticeable percentage of malicious users to evaluate the resilient to attacks of the models
Participation level at each iteration (when high)	50%	Half of the users provide recommendations at each iteration, which feeds the system with many feedbacks when this condition happens
Participation level at each iteration (when low)	5%	Only a small portion of users participate, which makes the system to not have many recommendations under this condition
Average quality of the service [0…1]	0.6	The service is slightly good
Change on the quality of service [No, Rarely, Often]	Often	The quality of service fluctuates. This decreases the accuracy of those models which do not adapt quickly

Table 2
Inference rules defined for the conducted experiments.

		SC_1	SC_2	SC_3	PM_1	PM_2	PM_3
RCE_1	R_1^1	M	M	L	H	H	L
	R_2^1	VH	VH	L	H	H	M
	R_3^1	L	L	H	L	M	M
RCE_2	R_1^2	M	M	L	H	H	H
	R_2^2	H	H	M	H	H	VH
	R_3^2	L	L	M	M	M	L
RCE_3	R_1^3	L	L	M	L	VL	H
	R_2^3	M	M	M	H	M	VH
	R_3^3	H	H	M	VH	VH	VH

RCE_1: Weighted Average
RCE_2: Preferences Weighted Average
RCE_3: Users Weighted Average
SC_1: Number of users
SC_2: Users' participation
SC_3: Percentage of malicious users
PM_1: Accuracy
PM_2: Adaptability
PM_3: Computer resources required
VL: Very low, L: Low, M: Medium,
H: High, VH: Very high.

value to the user, (iv) the user interacts with the service if it has reputation enough, and finally (v) the user provides a feedback, which actually feeds the reputation framework for future reputation computations. For end-users, providing feedbacks is a matter of rating the services based on their experience, whereas for IoT devices the rating could be based on pre-defined factors expected out of the service (e.g. response time, outcome stability, signal strength, etc.).

The tool performs several iterations of the described phases, showing relevant log information about the behavior of the scenario, such as the reputation values given by the reputation computation engine, the feedbacks provided by each user, the quality of the offered service, etc. So the accuracy of the reputation computation engine could be visualized and analyzed.

For this testbed we have extended the tool in such a way that the reputation service could deploy the proposed dynamic selector, named *Dynamic computation engine*, and use the most appropriate reputation computation engine at each moment. The selection of the reputation computation engine is based on the specific set of rules defined for each one of them, as shown in Table 2, whereas the ideal performance measurements to pursue are maximizing the accuracy and adaptability and minimizing the computer resources required ($PM_1 =$ VH, $PM_2 =$ VH, $PM_3 =$ VL).

Note that the five used tags, going from very low (VL) to very high (VH), correspond to five fuzzy sets which are proportionally split along the whole range of each system condition and performance measurement.

The tool has been also extended to simulate different system conditions for our tests. The system conditions are parametrized and their values are monitorized in such a way that when one of these values changes, the rules are analyzed to determine whether there is a more suitable candidate, and hence a swapping of the active reputation computation engine needs to be undertaken.

The implementation of the *Dynamic computation engine* within the simulator includes the smooth transition, where outputs from two different reputation computation engine are taken after swapping the active one. For this testbed, the transition time takes 7 iterations, and the weight of Eq. (1), which defines how the reputation values are considered during the transition period (Section 4), takes the values $\omega_i = \frac{6}{6}, \frac{5}{6}, \ldots, \frac{0}{6}$, whereas $\omega_k = \frac{0}{6}, \frac{1}{6}, \ldots, \frac{6}{6}$ in the respective iterations.

5.3. Outcomes

In order to compare the proposed solution with traditional reputation models, we set up several scenarios, aimed to model common IoT systems, including malicious users participating and changing both the conditions of the system and the quality of the service on-the-fly, making it harder for the reputation system to provide accurate reputation values unless they adapt quickly to the current circumstances.

To summarize the results obtained from the conducted experiments, we present the following example test which better illustrates them. In this test, we have set up all the reputation computation engines previously described working in parallel, but independently, with the same system conditions and simulation settings. That is, the tool simulates the same scenario with identical sequential phases for each of the reputation computation engines and obtains the outcomes they produce, such as the reputation values they compute. Comparing the computed reputation values with the real quality of service, the accuracy is computed, going from 0% (the computation engine provides extremely incorrect reputation values) to 100% (the engine always guesses perfectly the behavior of the service).

For this test, the simulated scenario starts with favorable system conditions (i.e. high amount of users, unlimited resources, high users' participation, etc.). At a certain moment, we simulate an abrupt decrease in the users' participation, remaining low for a while, and then it increases again.

This could be the case where, for instance, a complex IoT network has been defined and there are many nodes using and providing recommendations about a given service. However, in a certain moment, the topology of the network changes in such a way that a large percentage of the nodes that use the service cannot access it (i.e. user participation decreases) until they discover a new network path to access this service again (i.e. user participation increases again). Fig. 4 shows the accuracy of the computed reputation values of the different reputation computation engines along the time.

We can observe that, when having favorable conditions, the User Weighted Average computation engine provides higher accuracy when computing the reputation values (around 95%) than the Preferences Weighted Average engine (around 90%) and the

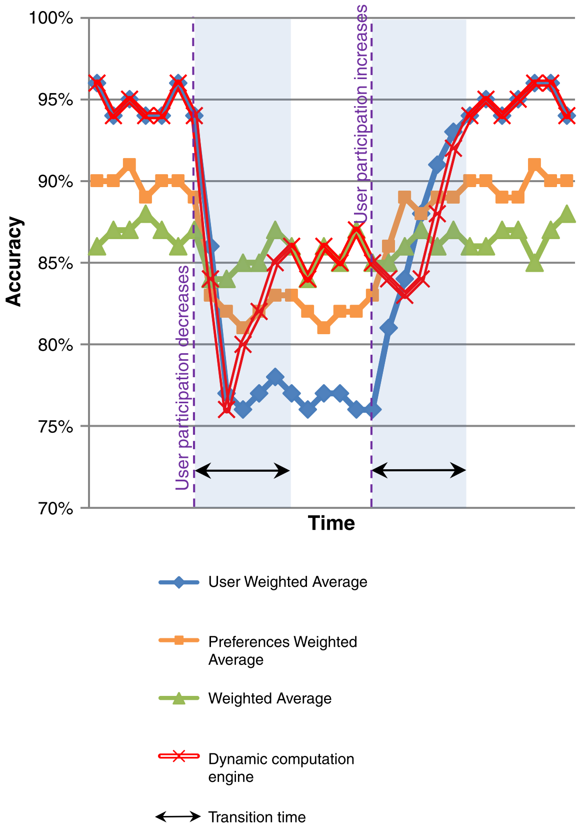


Fig. 4. Accuracy obtained from different RCEs in a dynamic scenario.

Weighted Average (around 87%). The dynamic selector has the User Weighted Average computation engine activated for such system conditions, hence providing identical accuracy to that engine.

Nevertheless, when the user participation decreases, the accuracy of both the User Weighted Average and Preferences Weighted Average computation engines notably decreases (around 77% and 82% respectively), since they do not work well with low users' participation. However, the accuracy of the Weighted Average computation engine is slightly different (around 85%), being the latter the computation engine which now provides higher accuracy, as it is not much affected by the users' participation.

When the dynamic selector detects the reduction on the users' participation, it activates the Weighted Average computation engine and triggers the smooth transition process. Then, our model starts to receive outcomes also from the recently activated computation engine and, little by little, gives less relevance to the outcomes of the User Weighted Average computation engine. Once the transition process is over (i.e. the User Weighted Average engine becomes completely idle and our model just receives outputs from the Weighted Average engine), it reaches an accuracy of around 85%. Note that in case the swapping had not occurred, the accuracy would have remained on around 77%.

When the users' participation increases later on, the accuracy of the User Weighted Average and Preferences Weighted average computation engines increase, being the former the computation engine with higher accuracy again (around 95%). As soon as the dynamic selector detects that the users' participation is high, it selects and activates the User Weighted Average. After the smooth transition process, our model also reaches an accuracy of around 95%. Overall, it can be seen that our model gets higher accuracy than the rest of the model working independently, since it tends to select the reputation computation engine which provides better outcomes when the system conditions change.

Observing Fig. 4, one might think that the dynamic selector should skip the transition time, and directly apply the new reputation computation engine once it detects that another computation engine would produce better outcomes. To avoid any confusion, it is worth mentioning that the figure shows the accuracy of the computation engines working in parallel, as if they all were working all the time. However, in the dynamic selector solution only the active computation engine is working, while the rest remain in an idle status. That is due to the fact that having all the reputation computation engines active would consume a huge amount of resources.

When the dynamic selector activates a computation engine, it requires a bootstrapping period until it begins to produce reliable outcomes. In other words, an idle computation engine does not start producing accurate reputation values as soon as it is activated, but it requires some time getting inputs to adjust weights and other parameters to stabilize its output, or to update those weights and parameters. For instance, in case they belong to a previous period when such reputation computation engine was active and therefore their weights and parameters are out-of-date and hence inaccurate.

In the following, we present a test performed to analyze the behavior of the dynamic selector when targeting the smooth transition process and when not. Based on the previous example, where the users' participation decreases at a certain moment and then increases again, we show in Fig. 5 the accuracy of the computation engines activated by the dynamic selector, together with the actual accuracy of the solution. In this case, we can ascertain that the reputation computation engines do not provide high accuracy for a while after being activated.

Fig. 5a shows the accuracy along the time using the smooth transition process. Even though the accuracy of the Weighted Average computation engine when activated is around 56%, in this figure we can observe that the output of the dynamic selector solution remains above 76%, since it is still getting values from the User Weighted Average computation engine.

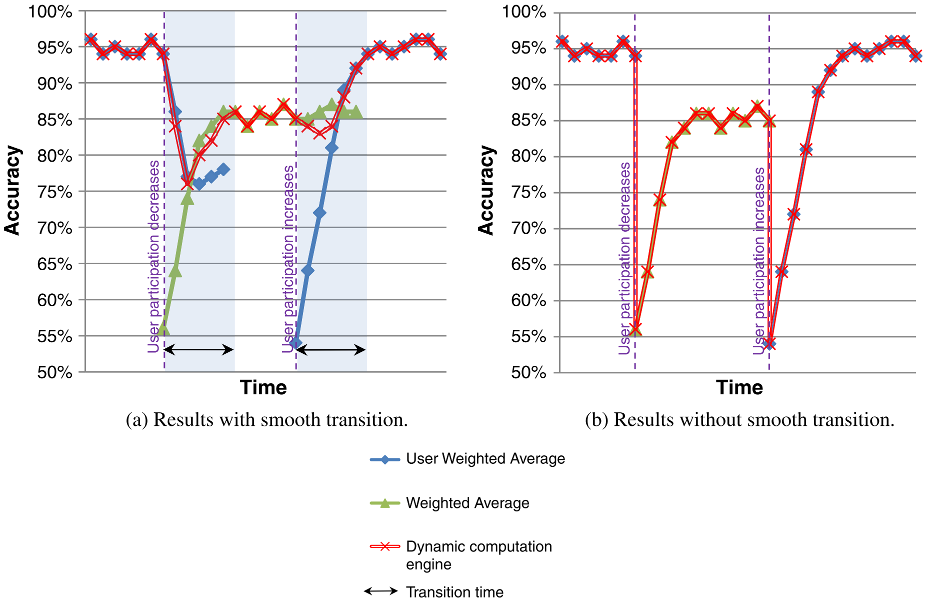
Furthermore, when the users' participation increases and the User Weighted Average computation engine is activated again, it starts producing an accuracy around 54%, but the dynamic selector remains above 83% since it is using the outcomes from the Weighted Average engine for a while, so the recently activated computation engine could finish the bootstrapping and start producing accurate outcomes.

Fig. 5b shows the opposite case, where the outputs of the reputation computation engines are fully taken into account as soon as they are activated, omitting the transition period. This figure shows how the accuracy of the dynamic selector drastically decreases when a change in the active computation engine is performed.

In the first change, the accuracy suddenly drops from 94% to 56%, thus making the reputation model very unreliable during the bootstrapping period. Something similar happens in the second change, where the accuracy drops again from 85% to 54%. In general, Fig. 5a properly sums up how costly the swapping between reputation computation engines would be, in terms of accuracy, without having the transition time.

6. Discussion

Many current reputation management systems rely on the use of a centralized entity or repository to collect feedbacks in order to provide reputation information of a service. For instance, there are many web services, such as eBay or Amazon [32], which use centralized reputation systems to rate the offered products. However, in IoT environments, where distributed and dynamic scenarios take place and each entity could act as consumer



(a) Results with smooth transition.

(b) Results without smooth transition.

Fig. 5. Transition between reputation computation engines.

and provider at the same time, other alternatives need to be deployed [21].

Reputation management systems have been successfully applied to different dynamic contexts, such as P2P [4,22,33,29,34], wireless sensor networks [2,35–37] or vehicular ad hoc networks [6,38] amongst others. Reputation management systems are usually focused on facing specific challenges, and designed to be applied on certain kinds of scenarios. For instance, some of them deploy sophisticated mechanism in order to avoid malicious recommenders (which are willing to increase or decrease the reputation of a service) or to provide personalized reputation values [23]. On the other hand, there are other models aimed to be as light as possible, in order to be deployed on low capabilities devices or to save batteries consumption [18].

Due to the vast amount of reputation models, it is hard for system designers or administrators to analyze the intrinsic features of each reputation model to figure out which one would produce the best outcomes or whether they would fulfil the performance requirements, such as the desirable accuracy, the expected computational time, etc. Furthermore, it constitutes a cumbersome task for them to decide which one fits better with their scenarios. Additionally, in IoT environments, the conditions of the scenarios, such as the number of IoT devices, their participation, capabilities, etc., would change along the time, and it would be even tougher to swap the reputation model once deployed.

EigenTrust [39] is a well-known reference within the reputation management field, which could be applied on IoT. It is characterized by the assignment of a trust value to each peer in a P2P system, which is based on the satisfaction that the peers have of each other after interacting between them. However, it is vulnerable to attacks on those IoT scenarios where an adversary could set several nodes trying to drive down the reputation of a reliable service.

Bandwidth is other relevant aspect that has to be carefully taken into consideration when choosing or designing a reputation model for an IoT scenario. [5] introduces mobile reputation agents, aimed to deal with those scenarios where the devices have high constraints in memory and computation. They tend to reduce the overhead in terms of extra messages and time delay, although sacrifices performance. In this way, even though other reputation models could produce better results, this model would be a better choice for low-consuming IoT environments. Furthermore, when the reputation is computed, this model takes into account the time when the feedback was received, in order to give higher relevance to the most recent feedbacks.

Oppositely, it could also be the case where computation and consumption capabilities are not critical aspects for some IoT scenarios. For those scenarios, some reputation models focused on cloud computing environments could be applied. [40] presents a reputation approach which computes opinions using subjective logic operators. This mechanism requires heavier computation and monitoring several aspects to compute the reputation values, which could make the system unsuitable for other IoT environments where the resources are more limited.

There are models in the literature directly oriented to face some IoT challenges. [41] proposed a trust management model, named TRM-IoT, based on fuzzy reputation for IoT. Nevertheless, this model is focused on the specific IoT scenario of wireless sensors networks. In this sense, this model takes into account a limited set of metrics to compute the reputation, such as package delivery ratio or energy consumption.

[42] designs an adaptive trust management protocol for social IoT, which are aimed to allow objects to have their own social networks. In this proposal, communities of interest are distributively and dynamically set up, in such a way that each node only updates trust towards others of its interest. Even though this model achieves dynamic adaptability, it is not clear how this model would be applicable in other scenarios where the social IoT paradigm is not being applied.

Several other reputation models are analyzed in [14] according to the fulfilment of common requirements, whereas a set of common security threats in trust and reputation management solutions and how they affect to several of the reputation models is presented in [43]. From these works it is straightforward to conclude that there is not a model that perfectly fits on every IoT scenario, due to scalability and a high variety of relationship among IoT entities, or that handles every security issue.

In addition, [15] describes a set of reputation computation engines, which take into consideration different aspects to compute reputation values, and hence with different levels of complexity. For example, whether the reputation computation engine takes into account users' preferences to compute customized reputation values, or whether they assign weights to the users to avoid malicious feedbacks. After analyzing their behavior against different system conditions and performance measurements, this work emphasizes the necessity of a system which selects the most suitable reputation computation engine depending on the current conditions of the environment which, to the best of our knowledge, is still missing.

7. Conclusions and future work

IoT introduces a world where a vast amount of heterogeneous devices communicate with each other, able to form complex and dynamic organizational structures. Along with this concept, the potential for new applications explodes, making the trust management aspect even more challenging. Users on the IoT need to decide whether or not they can trust a service or a device in such a flexible environment. Moreover, devices and services need to know whether they can trust each others.

Reputation management systems have been proposed in different contexts and scenarios to handle trust, especially in dynamic environments where mechanisms based on static trust relationships are no longer an option. Reputation management systems collect recommendations from different sources about an entity, usually based on past experiences, trying to predict its behavior and aiding in the decision on whether this entity is sufficiently trustworthy or not.

Even though some reputation management mechanisms have been conceived for dynamic environments, they usually offer rigid and inflexible mechanisms to compute reputation scores, which impedes their dynamic adaptation to the current circumstances in the system where they are deployed. At most, they provide certain parameters which are configurable or tunable. In the end, system administrators or designers have to manually select the most appropriate reputation model for their scenario, and shape this model to their needs, making very tough swapping it in the future.

In this document we have presented a dynamic mechanism able to select the most suitable reputation computation engine on-the-fly, in a flexible way. The selection is performed according to the current conditions of the system and the desired performance measurements, which are monitored by the system.

The proposed reputation selection mechanism is able to determine whether an idle reputation computation engine could provide better outcomes than the active one, and consequently exchange the active reputation computation engine. The transition between reputation computation engines is smoothly performed, in order to handle the bootstrapping period required to initialize and stabilize the parameters of the recently activated one.

Finally, a set of experiments have been conducted in order to validate the suitability of the proposed solution. It has been proved that this approach could output more accurate reputation values in the long term than those computed by traditional reputation models, where only one reputation computation engine is working. Besides, the necessity of performing a smooth transition between the current reputation engine and the new selected active one has been analyzed.

As future work, we foresee an ongoing research line to propose this mechanism in a standardization body, to allow an easy integration of reputation computation engines within the selection mechanism. As we have mentioned in this document, reputation models mainly differ on the reputation computation engines they implement. We were able to extract various reputation computation engines belonging to different models in order to validate this proposal. However, it would be useful to rely on a set of standard documents, presenting best practices and relevant use cases, which could be taken as reference by reputation model designers to properly isolate the reputation computation engines, making interoperability easier.

An additional research line coming from this work is to enhance the framework to help administrators in the process of defining the inference rules. To analyze the different reputation models beforehand is a tough task and prone to mistakes, which would limit the effectiveness of the framework. To prevent that, we are working on making the inference rule definition auto-adaptive. The idea behind is that the framework monitors and analyzes the accuracy of the reputation computation engine in use to be able to adapt or define inference rules used in the selection process. It would be able to learn if the reputation computation engine is behaving as expected and adapt the inference accordingly. In this way, if a reputation computation engine is not behaving as expected, new rules would be defined to prevent it from being selected under the current system conditions.

Acknowledgment

This research work has been partially funded by the Next-Generation Hybrid Broadcast Broadband (HBB-NEXT) EU research project, FP7-ICT-2011.1.5, Grant No.: 287848.

References

[1] M. Momani, S. Challa, Survey of trust models in different network domains, Int. J. Ad hoc, Sensor & Ubiquitous Comput. 1 (2010) 1–19.
[2] S.K. Bansal, A. Bansal, M. Blake, Trust-based dynamic web service composition using social network analysis, in: IEEE International Workshop on Business Applications for Social Network Analysis, BASNA 2010, pp. 1–8.
[3] C. Hu, J. Liu, Web services composition approach based on trust computing mode, in: 5th IEEE Asia–Pacific Services Computing Conference, APSCC 2010, pp. 663–670.
[4] F. Gómez Mármol, G. Martínez Pérez, State of the art in trust and reputation models in P2P networks, in: X. Shen, H. Yu, J. Buford, M. Akon (Eds.), Handbook of Peer-to-Peer Networking, Springer, US, 2010, pp. 761–784.
[5] A. Boukerche, L. Xu, K. El-Khatib, Trust-based security for wireless ad hoc and sensor networks, Comput. Commun. 30 (2007) 2413–2427.
[6] N.-W. Lo, H.-C. Tsai, A reputation system for traffic safety event on vehicular ad hoc networks, EURASIP J. Wirel. Commun. Netw. 2009 (2009) 9:1–9:2.
[7] S. Wang, L. Zhang, S. Wang, X. Qiu, A cloud-based trust model for evaluating quality of web services, J. Comput. Sci. Tech. 25 (2010) 1130–1142.
[8] G. Dólera Tormo, G. López Millán, G. Martínez Pérez, Definition of an advanced identity management infrastructure, Int. J. Inf. Secur. 12 (2013) 173–200.
[9] C. Fung, J. Zhang, I. Aib, R. Boutaba, Trust management and admission control for host-based collaborative intrusion detection, J. Netw. Syst. Manage. 19 (2010) 257–277.
[10] M. Gil Pérez, F. Gómez Mármol, G. Martínez Pérez, A.F. Skarmeta Gómez, RepCIDN: a reputation-based collaborative intrusion detection network to lessen the impact of malicious alarms, J. Netw. Syst. Manage. 21 (2013) 128–167.
[11] L. Atzori, A. Iera, G. Morabito, The Internet of Things: a survey, Comput. Netw. 54 (2010) 2787–2805.
[12] C. Medaglia, A. Serbanati, An overview of privacy and security issues in the Internet of Things, in: D. Giusto, A. Iera, G. Morabito, L. Atzori (Eds.), The Internet of Things, Springer, New York, 2010, pp. 389–395.
[13] G. Kortuem, F. Kawsar, D. Fitton, V. Sundramoorthy, Smart objects as building blocks for the Internet of Things, IEEE Internet Comput. 14 (2010) 44–51.
[14] F. Gómez Mármol, G. Martínez Pérez, Towards pre-standardization of trust and reputation models for distributed and heterogeneous systems, Comput. Stand. Interfaces 32 (2010) 185–196.
[15] G. Dólera Tormo, F. Gómez Mármol, G. Martínez Pérez, Towards the integration of reputation management in OpenID, Comput. Stand. Interfaces 36 (2014) 438–453.
[16] G. Dólera Tormo, F. Gómez Mármol, System and method for determining a reputation mechanism, Patent WO2013117224 (A1), 15/08/2013.
[17] L.A. Zadeh, Fuzzy sets, Inf. Control 8 (1965) 338–353.
[18] J. Yicka, B. Mukherjeea, D. Ghosal, A survey on sensor networks, Comput. Netw. 52 (2008) 2292–2330.
[19] H. Takabi, J.B.D. Joshi, G.-J. Ahn, Security and privacy challenges in cloud computing environments, IEEE Secur. Priv. 8 (2010) 24–31.
[20] L. Tan, N. Wang, Future internet: The Internet of Things, in: Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on, vol. 5, pp. V5–376–V5–380.
[21] R. Roman, P. Najera, J. Lopez, Securing the Internet of Things, Computer 44 (2011) 51–58.
[22] F. Gómez Mármol, G. Martínez Pérez, A.F. Gómez Skarmeta, TACS, a trust model for P2P networks, Wirel. Pers. Commun. 51 (2009) 153–164.

[23] H.K. Kim, J.K. Kim, Y.U. Ryu, Personalized recommendation over a customer network for ubiquitous shopping, IEEE Trans. Serv. Comput. 2 (2009) 140–151.

[24] J. Han, K. Micheline, Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.

[25] J.S.R. Jang, C.T. Sun, Functional equivalence between radial basis function networks and fuzzy inference systems, IEEE Trans. Neural Netw. 4 (1993) 156–159.

[26] W. Pedrycz, F. Gomide, An Introduction to Fuzzy Sets: Analysis and Design, The MIT Press, Cambridge, Masssachusetts, USA, 1998.

[27] L.A. Zadeh, From Computing with Numbers to Computing with Words—from Manipulation of Measurements to Manipulation of Perceptions, vol. 573, AIP, 2001, pp. 36–58.

[28] J.V. de Oliveira, Semantic constraints for membership function optimization, IEEE Trans. Syst. Man Cybern. A 29 (1999) 128–138.

[29] C. Huang, H. Hu, Z. Wang, A dynamic trust model based on feedback control mechanism for P2P applications, in: Autonomic and Trusted Computing, in: LNCS, vol. 4158, Springer, Wuhan, China, 2006, pp. 312–321.

[30] G. Dólera Tormo, F. Gómez Mármol, G. Martínez Pérez, ROMEO: ReputatiOn Model Enhancing OpenID Simulator, arXiv:1405.7831 [cs.CR], 2014.

[31] R. Chen, X. Chao, L. Tang, J. Hu, Z. Chen, CuboidTrust: a global reputation-based trust model in peer-to-peer networks, in: Autonomic and Trusted Computing, 4th International Conference, ATC 2007, in: LNCS, vol. 4610, Springer, Hong Kong, China, 2007, pp. 203–215.

[32] P. Resnick, R. Zeckhauser, J. Swanson, K. Lockwood, The value of reputation on eBay: a controlled experiment, Exp. Econ. 9 (2006) 79–101.

[33] Y. Wang, Y. Tao, P. Yu, F. Xu, J. Lu, A trust evolution model for P2P networks, in: Autonomic and Trusted Computing, 4th International Conference, ATC 2007, in: LNCS, vol. 4610, Springer, Hong Kong, China, 2007, pp. 216–225.

[34] S. Marti, H. García-Molina, Identity crisis: anonymity vs reputation in P2P systems, in: Proceedings for the Third International Conference on Peer-to-Peer Computing, P2P 2003, Linköping, Sweden, pp. 134–141.

[35] C.-W. Hang, M.P. Singh, Selecting trustworthy service in service-oriented environments, in: The 12th AAMAS Workshop on Trust in Agent Societies, pp. 1–12.

[36] Z. Malik, A. Bouguettaya, Reputation bootstrapping for trust establishment among web services, IEEE Internet Comput. 13 (2009) 40–47.

[37] S. Paradesi, P. Doshi, S. Swaika, Integrating behavioral trust in web service compositions, in: Proceedings of the 2009 IEEE International Conference on Web Services, ICWS '09, pp. 453–460.

[38] A. Patwardhan, A. Joshi, T. Finin, Y. Yesha, A Data Intensive Reputation Management Scheme for Vehicular Ad Hoc Networks, in: Proceedings of the Second International Workshop on Vehicle-to-Vehicle Communications, pp. 1–8.

[39] S.D. Kamvar, M.T. Schlosser, H. García-Molina, The eigentrust algorithm for reputation management in P2P networks, in: Proceedings of the 12th International Conference on World Wide Web WWW '03, ACM, New York, NY, USA, 2003, pp. 640–651.

[40] P. Pawar, M. Rajarajan, S.K. Nair, A. Zisman, Trust model for optimized cloud services, in: Trust Management VI, Springer, 2012, pp. 97–112.

[41] D. Chen, G. Chang, D. Sun, J. Li, J. Jia, X. Wang, TRM-IoT: a trust management model based on fuzzy reputation for Internet of Things, Comput. Sci. Inf. Syst. 8 (2011) 1207–1228.

[42] F. Bao, I.-R. Chen, J. Guo, Scalable, adaptive and survivable trust management for community of interest based Internet of Things systems, in: Autonomous Decentralized Systems (ISADS), 2013 IEEE Eleventh International Symposium on, pp. 1–7.

[43] F. Gómez Mármol, G. Martínez Pérez, Security threats scenarios in trust and reputation models for distributed systems, Elsevier Comput. & Secur. 28 (2009) 545–556.

Ginés Dólera Tormo is pursuing his Ph.D. at the University of Murcia. His research interests include authorization, authentication and identity management, user-centric technologies and trust management. He received an M.Sc. in computer science from the University of Murcia, Spain.

Félix Gómez Mármol is a senior researcher in the security group at NEC Laboratories Europe, Heidelberg, Germany. His research interests include authorization, authentication and trust management in distributed and heterogeneous systems, security management in mobile devices and design and implementation of security solutions for mobile and heterogeneous environments. He received his M.Sc. and Ph.D. in computer engineering from the University of Murcia, Spain.

Gregorio Martńez Pérez is an associate professor in the Department of Information and Communications Engineering of the University of Murcia. His research interests include security, privacy and management of IP-based communication networks. He received his M.Sc. and Ph.D. in computer engineering from the University of Murcia.

6

Identity Management: In privacy we trust. Bridging the trust gap in e-Health environments

Title:	Identity Management: In privacy we trust. Bridging the trust gap in e-Health environments
Authors:	Ginés Dólera Tormo, Félix Gómez Mármol, Joao Girao, Gregorio Martínez Pérez
Type:	Journal
Journal:	IEEE Security & Privacy, Special Issue on Health IT Security and Privacy
Publisher:	Elsevier
Volume:	11
Number:	6
Pages:	34-41
Year:	2013
DOI:	`http://dx.doi.org/10.1109/MSP.2013.80`
State:	Published

Table 6: Identity Management: In privacy we trust. Bridging the trust gap in e-Health environments

Identity Management—In Privacy We Trust:

Bridging the Trust Gap in eHealth Environments

Ginés Dólera Tormo, Félix Gómez Mármol, and Joao Girao | NEC Laboratories Europe
Gregorio Martínez Pérez | University of Murcia

Identity management solutions that control the data that users provide to individual healthcare services raise trust and privacy concerns, such as who owns user data, how to control its spread, and how to build trustworthy associations between care providers. Reputation systems can enhance eHealth systems by bridging the gap between strong contractual agreements and first-time domain exchanges.

Healthcare systems are making use of new technologies to offer their services to more people around the world. Enhanced services, such as remote patient monitoring and efficient emergency management, require users' information to be shared among multiple parties. Consumers of healthcare services appreciate the ease of use that identity management (IdM) solutions provide, such as single sign-on and the sharing of their data among services. However, these electronic transactions carry the risks of placing trust in the wrong provider and revealing private data to spammers and identity thieves.

IdM trust issues aren't new; however, the recent high proliferation of healthcare services, which adopt new requirements, necessitates new mechanisms to define and negotiate trust in a privacy-respecting manner. When care service providers (CSPs) want to cooperate with attribute providers (AttrPs) in an IdM context, the former require an exchange of user data to provide their services. Currently, many domains carry out these transactions securely using service-level agreements (SLAs)[1] and authentication, authorization, and accounting frameworks, but SLAs aren't always available or even easy to establish between domains.

In the past few years, much research addressed these drawbacks. Trust and reputation management has become a novel and effective way to tackle some of these security threats. Many models also effectively deal with concepts like trust and reputation over a wide range of environments, from P2P networks to wireless sensor networks or even multiagent systems.[2] Yet, privacy in trust and reputation management is often neglected. Privacy is particularly important because a user's recommendation about a CSP's trustworthiness directly ties into this user's identity in multiple care services. Moreover, users might avoid providing honest recommendations about a service or provider in fear of retaliation if their recommendations' privacy isn't preserved.[3] (For more information, see the "Related Work in Reputation Systems" sidebar.)

1540-7993/13/$31.00 © 2013 IEEE

In this article, we present an enhancement to the TRIMS (Trust and Reputation Model for Identity Management Systems) model, adding a mechanism that lets CSPs decide whether AttrPs are reliable enough to provide user attributes necessary to carry out a transaction.[4] In the same way, AttrPs can determine whether CSPs are reliable enough to obtain such attributes.

In TRIMS, the trustworthiness that an entity places on each recommender remains private.[4] We improve the privacy protection of the reputation data in the distributed system, ensuring that user feedback and recommenders' identifiers are known only by a trusted party and that the different parties' recommendations remain private.

Scenario Definition

As Figure 1 shows, we divide our scenario into several administrative domains. A healthcare service organization—for example, a clinic—can be considered as one domain since patient data exchange between services within this organization is subject to the providers' internal interfaces. However, when this organization wants to exchange data with a healthcare organization outside its domain—for example, an online pharmacy—the real trust and privacy problems manifest. The data exchanged can include the patient's health record, certain preferences, or even attributes that are part of the patient identity, such as name, address, and age. The service that receives patient data and that the patient accesses is the CSP (the online pharmacy in the aforementioned example), and the service that shares patient data is the AttrP (in this case, the clinic).

Identity providers (IdPs) are assumed to be trusted entities that manage identity information on behalf of users (and CSPs) and provide authentication assertions to other providers. In our approach, we use reputation to determine a domain's trust level, and IdPs act as recommendation aggregators—that is, they collect users' and CSPs' opinions on each IdP and return a single aggregated value.

In the example in Figure 1, a user asks for a certain service (for instance, some medication) from CSP_B in domain B (the online pharmacy). CSP_B requires some information (for example, the patient's allergies) from $AttrP_A$ in domain A (the patient's clinic) to carry out the user's request. First, CSP_B checks whether it's had past experiences with $AttrP_A$. Second, it asks other users who have had past experiences with $AttrP_A$ about its behavior. Finally, CSP_B asks other CSPs who have had past interactions with $AttrP_A$ about their satisfaction with such interactions.

After CSP_B collects all this information, it assesses how trustworthy $AttrP_A$ is, according to several self-defined trust levels. Each domain can determine its own trust levels depending on its needs and on how confident it is about the information gathered. According to the computed trust level, CSP_B requests all, partial, or no information.

Related Work in Reputation Systems

Identity management (IdM) systems deal with authentication and authorization processes, enabling single sign-on and mechanisms to exchange user information between different entities. Trustworthy entities, such as identity providers (IdPs) and attribute providers (AttrPs), maintain users' identity information and deliver this information when required by a service provider (SP) that users want access to.

Classic and simplistic IdM systems are commonly based on static trust relationships, in which SPs, AttrPs, and IdPs are supposed to be reliable. For example, SPs can validate user attributes, which are usually required to provide a service or achieve an authorization decision, because they've been issued by a trusted party. However, this condition isn't valid in dynamic environments, in which service-level agreements might not be established easily. To build trust relationships in dynamic scenarios in the absence of a central authority, some authors have proposed reputation systems wherein trust in an entity is based on its past interactions. For example, AttributeTrust constitutes an authorization framework with a reputation system to evaluate trust in user-provided attributes.[1] Although this solution helps SPs to decide whether to perform a transaction, it doesn't address privacy issues.

As Paul Resnick and Richard Zeckhauser note in "Trust among Strangers in Internet Transactions: Empirical Analysis of Ebay's

cont. on p. 36

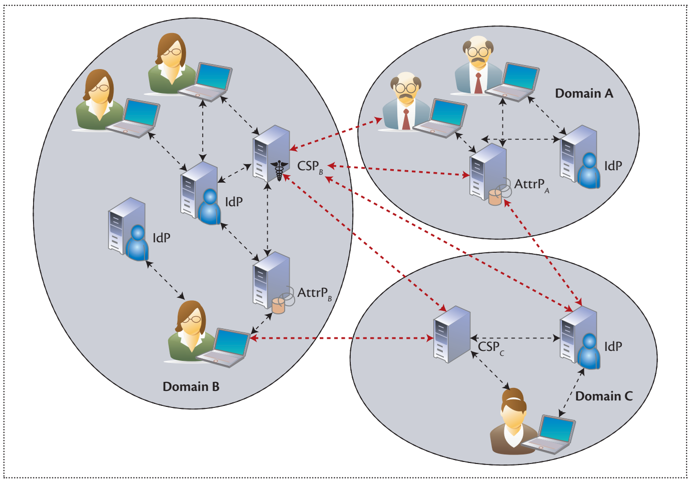


Figure 1. Scenario definition. Several domains comprising an ecosystem of care service providers (CSPs), attribute providers (AttrPs), and identity providers (IdPs) interact with one another to provide eHealth services to users while preserving their privacy.

Table A. Comparison of privacy-preserving reputation management solutions.

Article detailing solution	Reputation aggregator	Recommendation privacy	Weight privacy	User feedback privacy	Recommender privacy
"Towards Pre-standardization of Trust and Reputation Models for Distributed and Heterogeneous Systems"[9]	Recommenders	Recommendations are known by the querier	Each node's weights are known	Feedback is sent directly to the querier	Whole recommenders path is known by the querier
"TRIMS, a Privacy-Aware Trust and Reputation Model for Identity Management Systems"[8]	Querier	Recommendations are hidden	Weights are managed only by the querier	Feedback is known by the querier	Recommenders are known by the querier
Network and Traffic Engineering in Emerging Distributed Computing Applications[10]	Querier or trusted party	Recommendations are hidden	Weights are hidden	Unclear how the feedback is achieved in a private way	Recommenders are known by the querier and the trusted party
"Multi-party Trust Computation in Decentralized Environments"[7]	Querier	Recommendations are hidden	Weights are managed only by the querier	Feedback is known by the querier	Recommenders are known by the querier
"Schemes for Privately Computing Trust and Reputation"[5]	Recommenders	Recommendations are hidden	Weights are not applicable	Feedback is not collected	Recommenders are known by the querier and all the nodes involved
"Implementing Gentry's Fully-Homomorphic Encryption Scheme"[11]	Trusted party	Recommendations are known by the identity provider	Weights are hidden	Feedback is known by the querier	Recommenders are known only by the trusted party
Article at hand	Trusted party	Recommendations are hidden	Weights are hidden	Feedback is known only by the trusted party	Recommenders are known only by the trusted party

cont. from p. 35

Reputation System," if recommendations aren't provided in a privacy-preserving manner, recommenders of a reputation system might avoid providing honest feedback in fear of reprisals.[2] These systems must hide user feedback regarding the received service and keep secret the reliance that each entity has on other recommenders.

In turn, the work Rishab Nithyanand and Karthik Raman present in *Fuzzy Privacy Preserving Peer-to-Peer Reputation Management* enhances distributed reputation mechanisms to preserve peers' anonymity efficiently.[3] By using homomorphic encryption techniques,[4] this approach can compute peers' reputation without revealing the recommendations given by a specific peer or the weights used to model the impact of the different recommendations. (For more information, see "The Privacy Homomorphism Approach" sidebar.) Nevertheless, it could hardly be applied directly to IdM systems because user feedback could be sent to peers and user identity would hence be revealed.

In "Schemes for Privately Computing Trust and Reputation," Nurit Gal-Oz, Niv Gilboa, and Ehud Gudes present three protocols to compute reputation-based trust and prove them to be private against *semihonest*—that is, honest but curious—adversaries.[5] The authors assume that all entities involved follow the protocols correctly even though they would learn private information if

cont. on p. 37

Following the same approach, AttrP_A can collect recommendations about CSP_B to determine whether CSP_B is trustworthy enough to obtain the private information requested.

If CSP_B and AttrP_A trust each other enough to let the information exchange to occur (in this case, the patient's clinic informing the online pharmacy about the patient's allergies), then the service is provided to the user who requested it. This user will announce its satisfaction with that specific service to the CSP_B.

cont. from p. 36

possible. Following the protocols, neither recommendations nor weights to determine recommenders' reliability are revealed to other entities, but the reputation score could be computed according to these parameters. Furthermore, although the recommender's identity is known, the individual recommendations aren't. However, this solution doesn't allow the recommendation querier to adjust recommenders' trustworthiness because the querier doesn't know the individual recommendations.

Sebastian Ries and his colleagues handle this problem in "Learning Whom to Trust in a Privacy-Friendly Way."[6] They propose a privacy-preserving computation of trust that also enables the entities to learn about the recommenders' trustworthiness. Combining homomorphic encryption techniques with zero-knowledge proofs allows the querier to compute reputation values and calculate their accuracy to update recommenders' trustworthiness without revealing such information. However, in this solution, a user's feedback could be sent directly to the querier; thus, it's unclear how this feedback could be collected privately in an IdM context.

Likewise, in "Multi-party Trust Computation in Decentralized Environments," Tassos Dimitriou and Antonis Michalas present a protocol providing anonymous recommendations in reputation systems for decentralized environments.[7] Using *multiparty computation*—a cryptographic paradigm enabling different entities to jointly compute a function, without revealing their inputs to others—allows participants to cast their recommendations in a way that preserves the privacy of their recommendations; only the aggregated value is known. However, this solution doesn't deploy a mechanism to punish biased recommenders.

In "TRIMS, a Privacy-Aware Trust and Reputation Model for Identity Management Systems," Félix Gómez Mármol and his colleagues propose a trust and reputation model applied in IdM systems that can decide whether a Web service is reliable enough to receive users' attributes.[8] A trusted party can aggregate recommendations using homomorphic encryption techniques to keep secret the confidence placed on each recommender. Yet, the trusted party must know the recommendations given by the external entities, which could be considered sensitive information in some scenarios. In addition, user feedback is known by the querier, which can prevent users from providing accurate recommendations.

In the main text, we present a distributed reputation model to determine a specific entity's trustworthiness while preserving users' and entities' privacy. Although our model aggregates recommendations, it doesn't know users' or other entities' recommendation values about a specific entity. It also keeps secret the reliance that a specific entity gives to its recommenders, used to compute reputation values. Table A shows a summary of various reputation management solutions.

References

1. A. Mohan and D. Blough, "AttributeTrust—A Framework for Evaluating Trust in Aggregated Attributes via a Reputation System," *6th Ann. Conf. Privacy, Security, and Trust*, IEEE CS, 2008, pp. 201–212.
2. P. Resnick and R. Zeckhauser, "Trust among Strangers in Internet Transactions: Empirical Analysis of Ebay's Reputation System," *The Economics of the Internet and E-commerce*, vol. 11, 2002, p. 127.
3. R. Nithyanand and K. Raman, *Fuzzy Privacy Preserving Peer-to-Peer Reputation Management*, tech. report 2009/442, Cryptology ePrint Archive, 2009.
4. T. El Gamal, "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," *IEEE Trans. Information Theory*, vol. 31, no. 4, 1985, pp. 469–472.
5. N. Gal-Oz, N. Gilboa, and E. Gudes, "Schemes for Privately Computing Trust and Reputation," *Trust Management IV*, vol. 321, 2010, pp. 1–16.
6. S. Ries et al., "Learning Whom to Trust in a Privacy-Friendly Way," *Proc. 10th Int'l Conf. Trust, Security, and Privacy in Computing and Communications*, IEEE, 2011, pp. 214–225.
7. T. Dimitriou and A. Michalas, "Multi-party Trust Computation in Decentralized Environments," *Proc. 5th Int'l Conf. New Technologies, Mobility, and Security*, IEEE, 2012, pp. 1–5.
8. F. Gómez Mármol, J. Girao, and G. Martínez Pérez, "TRIMS, a Privacy-Aware Trust and Reputation Model for Identity Management Systems," *Elsevier Computer Networks J.*, vol. 54, no. 16, 2010, pp. 2899–2912.
9. F. Gómez Mármol and G. Martínez Pérez, "Towards Pre-standardization of Trust and Reputation Models for Distributed and Heterogeneous Systems," *Computer Standards & Interfaces*, Special Issue on Information and Communications Security, Privacy, and Trust: Standards and Regulations, vol. 32, no. 4, 2010, pp. 185–196.
10. J.H. Abawajy, *Network and Traffic Engineering in Emerging Distributed Computing Applications*, IGI Global, 2012, pp. 21–42.
11. C. Gentry and S. Halevi, "Implementing Gentry's Fully-Homomorphic Encryption Scheme," *Advances in Cryptology*, Springer-Verlag, 2011, pp. 129–148.

This feedback information can help CSP_B increase or decrease its trust in AttrP_A and also punish or reward the recommendations given in the previous step. A similar process is achieved providing AttrP_A's satisfaction with CSP_B so the former can update the latter's reputation.

In this context, reputation data is extremely sensitive because it reveals how users perceive the AttrP. AttrPs could use this information to prioritize users who rate the service trustworthy or refuse service to those who rate it low. In addition, the weights

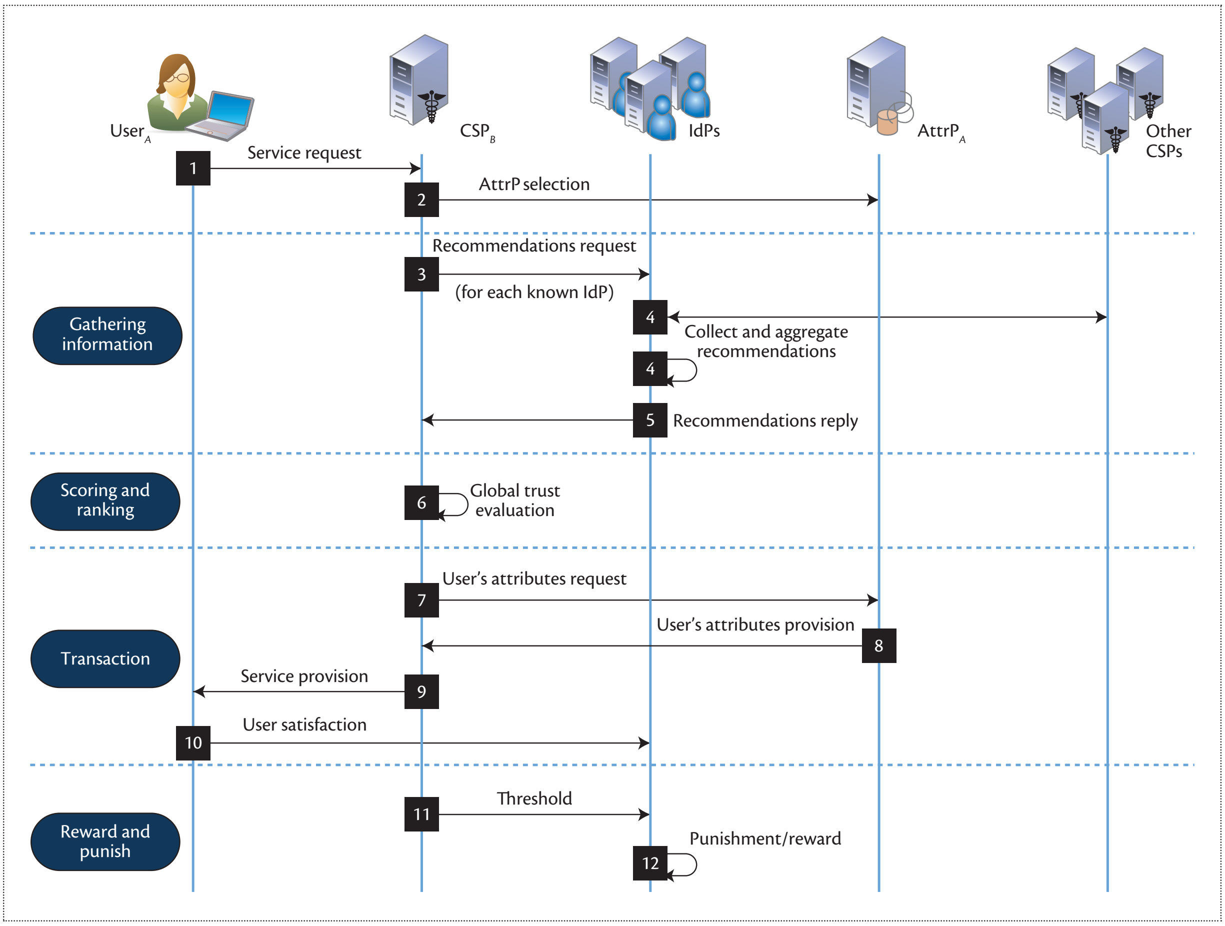


Figure 2. Sequence diagram of our model depicting the steps a CSP takes to assess an AttrP's trustworthiness in a privacy-preserving manner.

that CSPs give to each recommendation are equally sensitive because they could antagonize users and endanger the trust and reputation system's applicability and deployment.

Trust and Reputation Model Proposal

For a trust and reputation management system to be suitable for healthcare systems in an IdM context, it must provide a distributed way to collect, aggregate, and weigh recommendations from users who've engaged with an AttrP in the past. In addition, the recommendation values are sensitive and should be aggregated at the domains in which they're collected. Finally, the weights given to the individual recommendations should remain private.

Step by Step

Figure 2 depicts our model's main steps. We give an overview of the process here and then describe the steps in more detail.

1. A user (User$_A$) belonging to domain A requests a service from a CSP (CSP$_B$) belonging to domain B.
2. Through the user's IdP, CSP$_B$ selects an AttrP with the required user information (AttrP$_A$). Let's assume AttrP$_A$ belongs to the same domain as User$_A$.
3. CSP$_B$ requests all its known IdPs for their aggregated recommendations on AttrP$_A$.
4. Each IdP checks which of its users, or other CSPs it trusts, have interacted with AttrP$_A$ and aggregates these recommendations using a privacy homomorphism to preserve the recommendations' and weights' privacy.
5. The IdPs then return an aggregated recommendation to the CSP$_B$.
6. CSP$_B$ assesses its global trust about AttrP$_A$ and gauges the value using its internal trust levels.
7. CSP$_B$ then requests from AttrP$_A$ the user data associated with the selected trust level.
8. AttrP$_A$ provides the requested data. (Before

The Privacy Homomorphism Approach

Previous works have focused on maintaining privacy in the disclosure of user attributes. In this article, rather than focusing on attribute privacy or disclosure, we present identity management privacy in terms of disclosure of reputation information.

A homomorphic encryption scheme allows certain operations in encrypted data to reflect in the decrypted values. For instance, it could take two encrypted values as input, perform an addition operation without decrypting these values, and obtain encrypted results.

We apply this mechanism to the aggregation of recommendations from different domains to protect the weights the care service provider (CSP) applies to certain intermediate results and the values as they're aggregated hierarchically. The initial base value and the updates remain private even at the aggregators in the different domains, which is important to provide privacy when sharing sensitive information.

Although our approach offers a solution for data privacy, including aggregation, it doesn't provide a means to verify data authenticity or integrity. Here, we focus on privacy with a clear understanding that our approach must be combined with mechanisms to defend against attacks to data veracity.

Data gathering consists of additions and multiplications. We use a modified ElGamal encryption scheme to conduct these operations securely and privately. The ElGamal encryption scheme is a well-known multiplicative privacy homomorphism;[1] given the encryption of a message m, it's possible to construct a valid encryption of the message $t \cdot m$, for any t, without knowing either the private key or m. Furthermore, having the encryption of two messages m_1 and m_2, it could produce the encrypted message resulting from calculating $m_1 \cdot m_2$.

However, the ElGamal encryption scheme isn't additive homomorphic. We adapt the encryption scheme to work with an elliptic curve group, which can be used to get an additive homomorphic scheme. In this adaptation, each integer m within a finite range is mapped to a curve point M. A function $map(x)$ and its reverse function $rmap(x)$ are defined in a way that $map(m_1 + m_2) = map(m_1) + map(m_2) = M_1 + M_2$ holds. In this way, the privacy homomorphism properties are maintained throughout all operations on encrypted text.[2]

This encryption scheme's addition and scalar multiplication properties made it very suitable for our proposed trust mechanism. The scalar multiplication operation lets us multiply the recommendation values with the weight given by the CSP without revealing that value. By using the addition operation, we can add the values to calculate an aggregated sum without needing to decrypt the weighted values. The result can then be passed to the next level, in encrypted form, where the same operations can occur. In the end, the CSP can use the private key to decrypt the values.

There has been much discussion on whether homomorphic encryption is efficient and practical. Whereas the initial developments might not have been practical, current research is producing improved schemes that seem efficient. The performance of homomorphic encryption implementations is itself a question of interest. Indeed, it has been considered recently, for instance, in "Implementing Gentry's Fully-Homomorphic Encryption Scheme."[3] Furthermore, in "Can Homomorphic Encryption Be Practical?," Kristin Lauter and her colleagues show the extent to which current schemes can be used to compute functions of practical interest on encrypted data and implement and analyze an efficient homomorphic encryption scheme as proof of concept.[4]

References

1. T. El Gamal, "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," *IEEE Trans. Information Theory*, vol. 31, no. 4, 1985, pp. 469–472.
2. J.M. Adler et al., "Computational Details of the VoteHere Homomorphic Election System," *Proc. Int'l Conf. Theory and Application of Cryptology and Information Security* (ASIACRYPT 00), 2000.
3. C. Gentry and S. Halevi, "Implementing Gentry's Fully-Homomorphic Encryption Scheme," *Advances in Cryptology*, Springer-Verlag, 2011, pp. 129–148.
4. K. Lauter, M. Naehrig, and V. Vaikuntanathan, "Can Homomorphic Encryption Be Practical?," *Proc. 3rd ACM Workshop Cloud Computing Security Workshop*, ACM, 2011, pp. 113–124.

providing the data, AttrP_A might want to repeat the trust verification process, obtaining recommendations about CSP_B to estimate its trust level of CSP_B. For simplicity, we don't show this process.)

9. CSP_B delivers the service to the user.
10. The user provides feedback, securely and privately, to his or her IdP including his or her satisfaction with the received service.
11. Using a secure communication channel, CSP_B sends a threshold to every known IdP. This threshold represents the factor of punishment or reward to apply to each recommendation source.
12. Each IdP performs the corresponding punishment or reward to all recommenders, which depends on their recommendations' threshold and accuracy. User identity is not revealed at any stage.

Trust and Reputation Model Design

As suggested in "Towards Pre-standardization of Trust and Reputation Models for Distributed and Heterogeneous Systems,"[5] our trust and reputation model consists of four main steps, namely: collecting recommendations, scoring and ranking, service provisioning, and reward and punishment.

Collecting recommendations. We use three information sources—the CSP evaluating the AttrP, users, and other CSPs who've had experience with the AttrP. In the example in Figure 1, when CSP$_B$ computes its trust in AttrP$_A$, it checks whether it's had previous interactions with AttrP$_A$. If so, the last computed global trust value for AttrP$_A$ is taken as the direct trust.

Moreover, our model considers the trust that other users and CSPs have in AttrP$_A$. CSP$_B$ then needs to find the users and CSPs who have interacted with that AttrP$_A$ in the past. CSP$_B$ obtains this information from the different IdPs it trusts. These IdPs could be trusted by CSP$_B$ since they have a direct trust relationship established—for instance, if they're located in the same domain as CSP$_B$—or if indirect trust has been established—for instance, those IdPs are reliable by a trustworthy IdP.

Because all users and CSPs store their recommendations, we propose that each IdP store the weight given by CSP$_B$ to each of its users (and CSPs) but encrypt this with CSP$_B$'s public key, so the IdP can't unveil the weight values. Thus, each IdP computes the weighted aggregation of all its users' and CSPs' recommendations and gives it back to CSP$_B$, also encrypted with CSP$_B$'s public key. CSP$_B$ then decrypts that aggregation with its private key to obtain the weighted recommendation of all the users belonging to that IdP. To accomplish this, we need a special class of encryption functions $\mathfrak{E}$, called privacy homomorphisms,[6] which fulfill the following condition:

$$\Sigma_{i=1}^{n}\mathfrak{E}(\omega_{u_i})\cdot Rec_{u_i}=\mathfrak{E}\left(\Sigma_{i=1}^{n}\omega_{u_i}\cdot Rec_{u_i}\right), \qquad (1)$$

where ω_{u_i} is the weight given by CSP$_B$ to user u_i and Rec_{u_i} is the recommendation given by user u_i about AttrP$_A$. (For more information on privacy homomorphisms, see the related sidebar.)

Hence, CSP$_B$ can weight each user's recommendation individually, and each user's actual recommendation value is known only by its corresponding IdP, achieving a smart decoupling of information. The importance of these homomorphic schemes lies in the preservation of user recommendations' confidentiality and privacy. In fact, the CSP doesn't need to know each recommendation individually, but it needs to weight them.

Scoring and ranking. Once all the data related to the targeted AttrP$_A$ has been collected, our model computes a global trust value by aggregation. Thus, the users' trust in AttrP$_A$ is computed by a weighted sum of each user's recommendations, as Equation 1 shows. The trust that other CSPs have in AttrP$_A$ is similarly obtained. If the other CSPs don't want to release their recommendations to the IdP to maintain their privacy, they could receive the encrypted weight, compute Equation 1 themselves, and return the encrypted result; however, this would increase complexity.

Service provisioning. After computing the AttrP$_A$'s global trust value, CSP$_B$ must gauge the trust level. Each CSP can define its own trust levels. We suggest the use of fuzzy sets, whose linguistic labels enhance the model's interpretability.[4]

With fuzzy sets, each trust level has an associated amount or type of information that can be exchanged. To determine an AttrP's trust level from its global trust value, we need to know the values returned by the membership functions of every self-defined fuzzy set containing this value as an element. Once we have all those values, we select the trust level with a probability directly proportional to the value returned by its membership function. The flexibility and nondeterminism given by using fuzzy sets help the model to dynamically evolve over time.[7] Finally, CSP$_B$ requests from AttrP$_A$ the user attributes associated with the selected trust level.

Reward and punishment. Once the transaction has been carried out between CSP$_B$ and AttrP$_A$, a reward or punishment is applied to users and CSPs according to their recommendations' accuracy. Again, this process should be achieved while preserving the recommendations' and their weights' privacy.

To this end, CSP$_B$ sends to its associated IdPs a certain threshold that determines the factor of punishment or reward that should be applied to the recommenders. The CSP determines this threshold usually based on some preferences or on the provided services, then encrypts it with the CSP's public key. For instance, a CSP that offers critical services could establish a threshold that strongly punishes recommenders unless they provide highly accurate recommendations.

A given recommender's accuracy—and hence the amount of punishment or reward that should be applied—is proportional to the distance between the given recommendation and the satisfaction perceived by the user regarding the service provided by the CSP. That is, the closer those values are, the greater the reward, and the further those values are, the greater the punishment.

Although the current recommender weight and threshold values are encrypted using the CSP$_B$ public key, the IdP could compute the new weight associated with a recommender by using homomorphic encryption techniques, multiplying the current weight by a value representing the punishment or reward (that is, a value between 0 and 1 for punishment and more than 1 for reward). In turn, the punishment or reward value is calculated by multiplying the threshold by the accuracy.

Moreover, the evolution along the time of the weights given to each information source will depend on the source's accuracy. For an analysis of this reputation technique's accuracy, see "TRIMS, a Privacy-Aware Trust and Reputation Model for Identity Management Systems."[4]

Finally, the generation of new identities for newcomers should have an associated cost (not necessarily economic) to prevent a malicious user from creating a large amount of bogus identities and providing fake recommendations.

For immediate future work, we're considering the implementation and deployment of our model over a real scenario like those proposed by the Kantara Initiative (a nonprofit professional association dedicated to advancing technical and legal innovation related to digital identity management) as well as its proposition for standardization. Furthermore, although our approach offers a solution for data privacy, including aggregation, it doesn't provide a means to confirm data authenticity or integrity. In the future, we hope to add a mechanism to verify these characteristics in the aggregated encrypted recommendations. ■

References

1. J.H. Abawajy, *Network and Traffic Engineering in Emerging Distributed Computing Applications*, IGI Global, 2012, pp. 21–42.
2. A. Josang, R. Ismail, and C. Boyd, "A Survey of Trust and Reputation Systems for Online Service Provision," *Decision Support Systems*, vol. 43, no. 2, 2007, pp. 618–644.
3. P. Resnick and R. Zeckhauser, "Trust among Strangers in Internet Transactions: Empirical Analysis of Ebay's Reputation System," *Economics of the Internet and E-commerce*, vol. 11, 2002, p. 127.
4. F. Gómez Mármol, J. Girao, and G. Martínez Pérez, "TRIMS, a Privacy-Aware Trust and Reputation Model for Identity Management Systems," *Elsevier Computer Networks J.*, vol. 54, no. 16, 2010, pp. 2899–2912.
5. F. Gómez Mármol and G. Martínez Pérez, "Towards Pre-standardization of Trust and Reputation Models for Distributed and Heterogeneous Systems," *Computer Standards & Interfaces*, Special Issue on Information and Communications Security, Privacy, and Trust: Standards and Regulations, vol. 32, no. 4, 2010, pp. 185–196.
6. T. El Gamal, "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," *IEEE Trans. Information Theory*, vol. 31, no. 4, 1985, pp. 469–472.
7. P.-C. Cheng et al., "Fuzzy Multi-level Security: An Experiment on Quantified Risk-Adaptive Access Control," *Proc. IEEE Symp. Security and Privacy*, IEEE CS, 2007, pp. 222–230.

Ginés Dólera Tormo is a research scientist in the security group at NEC Laboratories Europe, Heidelberg, Germany. He is pursuing his PhD at the University of Murcia. His research interests include authorization, authentication, and identity management; user-centric technologies; and trust management. Dólera Tormo received an MSc in computer science from the University of Murcia. Contact him at gines.dolera@neclab.eu.

Félix Gómez Mármol is a senior researcher in the security group at NEC Laboratories Europe, Heidelberg, Germany. His research interests include authorization, authentication, and trust management in distributed and heterogeneous systems; security management in mobile devices; and design and implementation of security solutions for mobile and heterogeneous environments. Gómez Mármol received a PhD in computer engineering from the University of Murcia. Contact him at felix.gomez-marmol@neclab.eu.

Joao Girao is a group manager in the Security Group at NEC Laboratories Europe, Heidelberg, Germany. His research interests include security for networks and services and identity management. Girao received a diploma in computer and telematics engineering from the University of Aveiro, Portugal. He is a member of IEEE and the ACM. Contact him at joao.girao@neclab.eu.

Gregorio Martínez Pérez is an associate professor in the Department of Information and Communications Engineering at the University of Murcia. His research interests include security, privacy, and management of IP-based communication networks. Martínez Pérez received a PhD in computer engineering from the University of Murcia. Contact him at gregorio@um.es.

IT Professional
TECHNOLOGY SOLUTIONS FOR THE ENTERPRISE
www.computer.org/itpro

7

Towards privacy-preserving reputation management for hybrid broadcast broadband applications

Title:	Towards privacy-preserving reputation management for hybrid broadcast broadband applications
Authors:	Ginés Dólera Tormo, Félix Gómez Mármol, Gregorio Martínez Pérez
Type:	Journal
Journal:	Computers & Security
Publisher:	Elsevier
Year:	2014
Month:	September
State:	Accepted

Table 7: Towards privacy-preserving reputation management for hybrid broadcast broadband applications

Towards privacy-preserving reputation management for hybrid broadcast broadband applications

Ginés Dólera Tormo[b,*], Félix Gómez Mármol[a], Gregorio Martínez Pérez[b]

[a]NEC Europe Ltd. Kurfürsten-Anlage 36, 69115 Heidelberg, Germany
[b]Department of Information and Communications Engineering, University of Murcia, 30100 Murcia, Spain

Abstract

Hybrid Broadcast Broadband TV (HbbTV) is an industry standard aimed to provide a platform combining TV services with Internet services, using connected TVs and set-top boxes. It enables the possibility for vendors to offer applications directly to the users, introducing new entertainment services such as streaming of video on demand, games, social networking, etc. As a consequence, tons of applications are available for users to directly download and consume through the so-called application stores, despite the potential trust and security issues arising due to the decentralized nature of these environments.

Reputation management systems are usually deployed to handle trust in such dynamic scenarios, whereas they could also be used to evaluate and rate the applications from the users' point of view, and even provide customized rankings. Nevertheless, they require the application stores to know information related to the installed applications and the provided recommendations of the users, hence compromising their privacy.

In this paper we present a privacy-preserving reputation management framework to be integrated within the HbbTV context. We make use of identity management and extend homomorphic encryption techniques to avoid the application stores and other relying parties determining the recommendations provided by the users, yet being able to compute customized reputation values based on the similarity between users' recommendations.

Keywords: HbbTV, Reputation Management, Homomorphic Encryption, Identity Management, Privacy

1. Introduction

Hybrid Broadcast Broadband TV (HbbTV) is an initiative and an industry standard [1] aimed to combine television services delivered via broadcast with services delivered via broadband (e.g. Internet access). It enables a hybrid scenario between television and web content in order to offer enhanced services to the users through their television screens.

[*] The work presented in this paper was performed while working at NEC Laboratories Europe. Corresponding author. Tel.: +34 868 887646; fax: +34 868 884151.

Email addresses: ginesdt@um.es (Ginés Dólera Tormo), felix.gomez-marmol@neclab.eu (Félix Gómez Mármol), gregorio@um.es (Gregorio Martínez Pérez)

Preprint submitted to Computer & Security *August 30, 2014*

This standard provides the features and functionality required to deliver an interactive TV, allowing new entertainment services such as streaming of video on demand, interactive advertisement, access to customized information, games, applications, web surfing, social networking, etc.

The HbbTV initiative opens the door to new business opportunities, where lots of vendors are starting to offer their services, taking advantage of the rapid application development that HbbTV offers. Once an application is implemented following the specification, it runs on any device that supports the HbbTV standard, no matter who the hardware manufacturer is.

Similar to other contexts, one of its main advantages is the possibility to offer vendor applications directly to the users by means of an application store, also referred to as app-store or application marketplace [2, 3, 4, 5]. Users can browse through different categories, view information and reviews of the applications, purchase, download and install them on their device.

Furthermore, due to its decentralized nature and simplicity, there is a business opportunity for everyone with enough creativity and entrepreneurship. As a consequence, tons of applications are available for users to directly download and consume, which, on the other hand, potentially raises trust and security issues for the end users.

Application stores usually deploy mechanisms to avoid posting malicious applications, for instance using automatic analysis tools when a new application is received [6]. However, these mechanisms cannot guarantee hundred per cent of success and they are therefore complemented with more dynamic trust management solutions.

Reputation management systems have been widely spread and applied in order to handle trust in dynamic environments [7, 8, 9]. They have been proved to be effective on several contexts, such as P2P [10, 11] or service-oriented environments [12, 13], though, to the best of our knowledge, there are no trust and reputation models or solutions specifically applied to the field of HBB. The main goal of reputation management systems within HbbTV domain would be to evaluate the trustworthiness of the HBB applications offered to the users, in order to assist the latter in the decision of which applications to install and effectively consume.

However, the concepts of trust and reputation should not be mistaken. Whenever we are unable to rely on our own experience, we must fall back on recommendations and judgements, which we then use to guide us. Thus, reputation expresses the collective opinion, leading to trust or distrust.

Besides trust issues, reputation management could also be used to evaluate the applications from the users' point of view, based on their personal recommendations. The users could rate their installed applications to help other users to decide whether these applications are worth buying or installing. These ratings could be additionally used to establish some ranking or even to order the applications based on rated parameters, such as quality, functionality, ease of use, etc.

Some improvements have been achieved on reputation management systems in order to obtain better accuracy of the reputation values. They not only try to avoid malicious users, willing to increase or decrease the reputation of certain applications [14], but they also try to provide personalized reputation values [15]. For example, the application store could recommend different applications to different users depending on their interests. To that end, when computing the reputation for a given user, these mechanisms give more relevance to the recommendations provided by users with similar interests to a particular user. This similarity is usually computed based on user's preferences, their installed applications, or even on the rating given to installed applications [16].

Nevertheless, these improvements would require the application store to know all the information related to the installed applications and the provided recommendations of each of its users, which could be considered as a privacy lack in the HBB context. In fact, it has been studied that users may avoid providing honest recommendations about a service in fear of retaliation if recommendations cannot be provided in a privacy-preserving way [17].

In this paper we present a detailed description of a reputation framework to be integrated within the HBB context. The reputation framework is able to compute personalized reputation values, based on the similarity of the users' recommendations, but preserving the privacy of those users. In the proposed solution, we introduce identity providers [18] aimed to provide authentication functionalities whereas they hide the real identity of the users to the application store.

Moreover, even though identity providers are considered trusted parties, we make use of homomorphic encryption techniques [19, 20] to also hide the recommendations provided by the users to the identity providers. In this way, neither the application store nor the identity providers could determine the recommendations provided by a given user, yet they can compute personalized reputation values. In addition, since current homomorphic encryption techniques only have been proved to be efficient operating on encrypted bits, we present some algorithms aimed to extend that functionality so it could be used to compute natural numbers.

The remainder of the article is organized as follows. Section 2 introduces the terminology of the app ecosystem, whereas Section 3 describes threats and common solutions to be protected from them. Section 4 introduces reputation management systems and their common behaviour. Section 5 presents the proposed privacy-aware reputation management framework aimed to provide customized reputation values. Section 6 analyzes the proposed framework against common reputation and privacy attacks whereas Section 7 presents the results of the testbed performed to analyze the behavior of the proposed solution in terms of performance. Section 8 provides the main references and related work, and finally Section 9 describes the main conclusions derived from this work.

2. App Ecosystems: definitions and concepts

An *app* or application is a piece of software that extends the functionality of a user device. Within the HBB context, a user device refers mainly to set-top-boxes (STB). Before installing a specific app, end users can browse from their STB the app description and reputation (number of downloads, users votes and users reviews) provided by the application store that distributes it. Once they decide to install it, the app can be run on the STB and users can submit comments and complaints about it. A periodic check may allow the application store to update the app or even to remotely remove it from the STB if the app is considered insecure.

These interactions between end-users and generic application stores are shown on Figure 1. But the app ecosystem scene would not be complete without those previous interactions that take place between app developers and application stores, as shown on Figure 2.

App developers create new apps and update existing ones, but these apps are generally sold or distributed to end-users through the so called application stores. Thus, application stores could be described as managed repositories of third-parties' software [21].

Before accepting a new app from developers, application stores resort to defence mechanisms to check whether the app is suitable for inclusion or not. The fact that malware is introduced on user devices mainly via apps or application stores leads us, in Section 3, to review in more detail malware threats in app ecosystems and available mechanisms of defence.

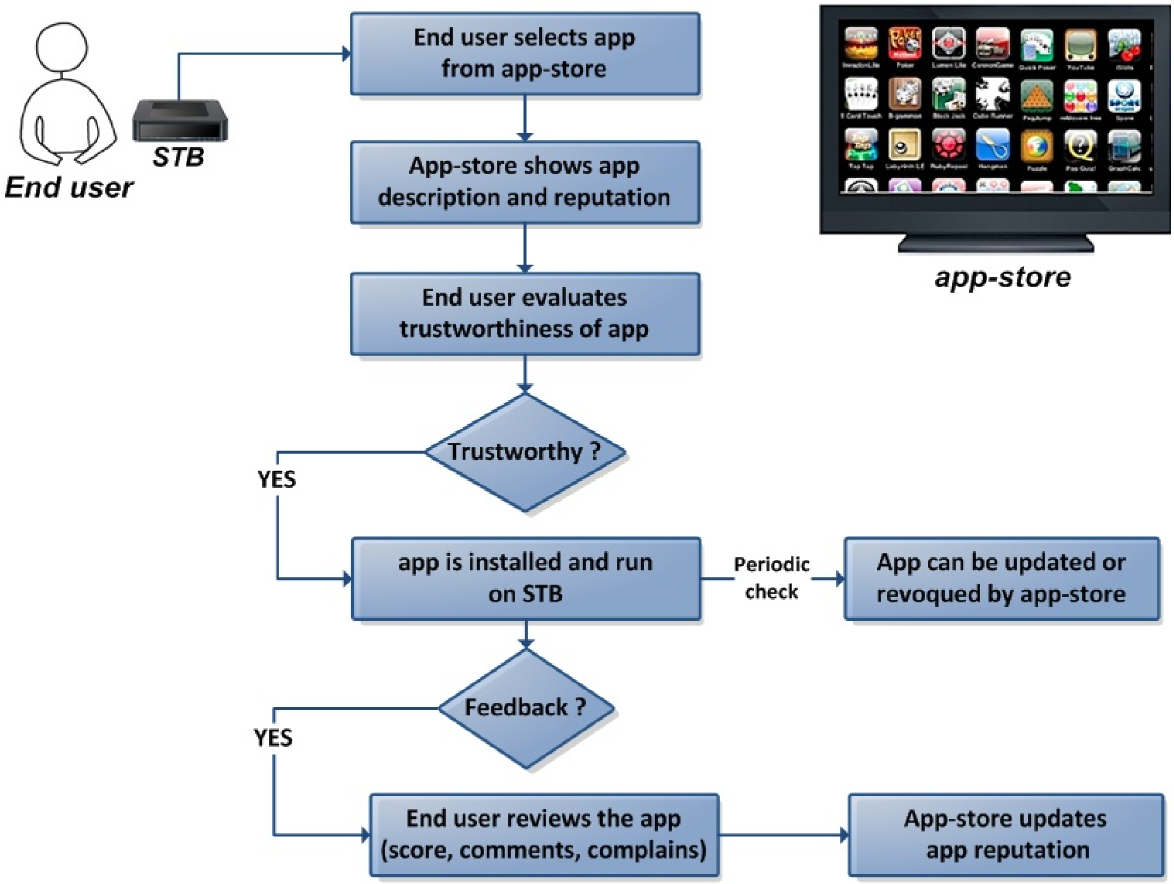

End user
STB
End user selects app from app-store
App-store shows app description and reputation
End user evaluates trustworthiness of app
Trustworthy ?
YES
app is installed and run on STB
Periodic check
App can be updated or revoqued by app-store
Feedback ?
YES
End user reviews the app (score, comments, complains)
App-store updates app reputation
app-store

Figure 1: Interaction between end users and application stores

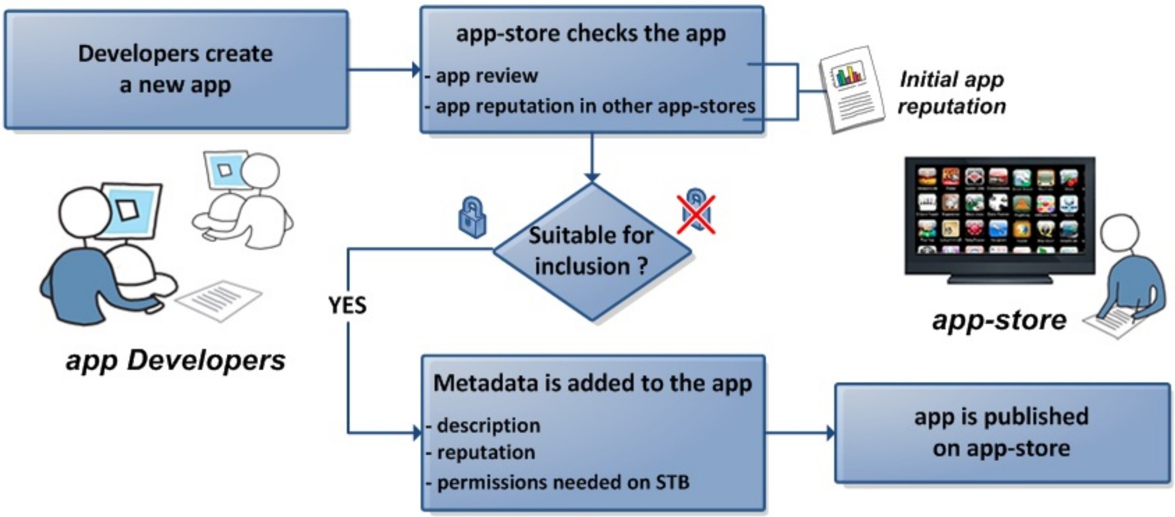

Developers create a new app
app-store checks the app
- app review
- app reputation in other app-stores
Initial app reputation
app Developers
Suitable for inclusion ?
YES
app-store
Metadata is added to the app
- description
- reputation
- permissions needed on STB
app is published on app-store

Figure 2: Interaction between app developers and application stores

Once an app is approved for inclusion in an application store, it is packaged by adding metadata such as a description or a list of permissions that the app needs on the user device. An early reputation for the new app is then built to help future users make an informed trust decision about it.

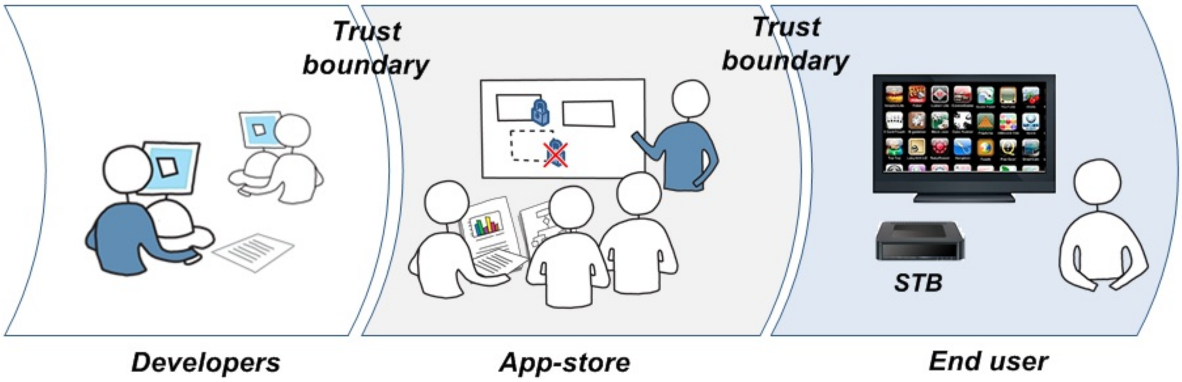


Figure 3: Description of an app-ecosystem

Gathering together the 3 previously described agents (i.e., app developers, application stores and app end users) we get the whole picture of an app ecosystem. As shown on Figure 3, two trust boundaries could be identified. They indicate the edges of control, on one hand between the developer and the application store (under control of different owners), and on the other hand between the application store and the STB (the latter under control of the user).

Thus, application stores are constantly dealing with trust decisions whenever they interact either with app developers or app end-users. It is for this reason that securing application stores is of great importance. Indeed, application stores generally deploy some defence mechanisms that enable them to provide end-users with vetted app distribution channels, to show the reputation of apps and developers, and to operate a revocation mechanism for malware and insecure app [4]. Nevertheless, it still cannot give 100% guarantee against malware attacks.

3. Threats Analysis and Defense

The high level goals of attackers of app ecosystems are to get malicious code on the user devices, and to keep that malicious code on there as long as possible. To reach these goals, they will try to directly sell or distribute malicious apps (e.g. Gemini [22], a trojan with botnet-capabilities that infected smartphone user in China by the end of 2010; or DroidDreams [23], a smartphone trojan hidden in look-like versions of popular apps piggybacking onto their reputation) or to exploit software vulnerabilities in existing popular apps (e.g., Zitmo malware [24, 25, 26], a smartphone Trojan designed to capture online banking SMS messages).

The fact that malware is most of the times introduced on user devices via apps or application stores, lead ENISA (European Network and Information Security Agency) to outline five lines of defence [27] that protect end-users from malware and insecure apps.

3.1. App review

Apps should be checked for security issues before they are posted on an application store and, hence, distributed to end-users. This vetting process [28] is performed by i) automatic analysis tools, which includes virus scanning, source code analysis against developer guidelines or running the apps against a number of test cases and, to some extent, ii) human reviews, although that raises some discussion in regards to scalability. Even though this process cannot guarantee 100% protection, it limits the introduction of malicious (or legitimate but insecure) apps in the app ecosystem.

3.2. Device security: Sandboxes

The user device (i.e., the set-top-box) should install and run apps in sandboxes to reduce the malware impact. Sandboxes are controlled environments within the user device where apps can be installed and run in isolation. In the sandbox [29], apps should only get a minimal set of privileges by default. The user should be explicitly asked for consent in case any additional privilege is required by the app, and should also be able to monitor the app activity within the sandbox. Nevertheless, in case the malware gets to break out of the app sandbox, it may be necessary to use removal tools, for instance a kill-switch.

3.3. App revocation: Kill-switch

App platforms should support remote removal of malware and insecure apps installed on end user devices. When an app is installed with a “kill-switch” and is later considered insecure, the application store that provided it is able to trigger the app revocation, remotely uninstall it and return the user device to a pre-install state. The device platform should be designed in such a way that it should not be possible for an app to get rid of the “kill-switch” element without being detected. Yet, the user should be properly informed and asked for consent before any revocation is performed.

3.4. Jails or walled gardens

Jails or walled gardens are an approach to app installation policies. That is to say, set-top-boxes could be restricted to only install apps from certain trusted application stores, or be configured to present the user clear warnings about installing apps from untrusted application stores. This is a crucial point, as other defense mechanism could become useless if users skip the warnings and start installing apps from untrustworthy sources. Furthermore, if jails are too restrictive, users are more tempted to jailbreak their devices (i.e. skip the restriction of their devices), which could break other security mechanisms, making them more vulnerable to other attacks.

3.5. Reputation mechanisms

In order to help users to choose trustworthy apps, application stores should show the reputation of apps (and app developers), i.e. history and track record based on download statistics as well as on users votes, comments and complaints.

Once a user downloads a certain app, she should be allowed to rate it and to give feedback on functionality and security relevant features (such as excessive permission requests). To increase scoring quality and to prevent Sybil attacks (where attackers create multiple pseudonymous identities to gain excessive influence and bias reputation results), more weight could be given to those votes from users who have a good reputation as raters.

However, the first time a new app is published on an application store and before getting any user feedback, its initial reputation is built up accordingly to the app security mechanisms (run in a sandbox on the user device, installed with or without a kill-switch, etc.) and taking into account its reputation in other trusted application stores. The latter would entail having unique identifiers for apps and signatures to allow referencing across different application stores.

Ideally, application stores should try to come to an industry-wide set of security principles to safeguard their end-users. However, they vary considerably from vendor to vendor in terms of the way they address malware and insecure apps [30]. A cross-platform reputation system that works across app stores seems unlikely to achieve. In fact, during the analysis of the different reputation models on use, it was evident that service providers are reluctant to provide details of their own reputation systems, as they are considered to be intellectual property that differentiate the service and provide business advantage. Additionally, exposing details of how a reputation system operates could reveal its vulnerabilities to fraud and manipulation.

Focusing on security issues, this individual development of different reputation systems increases the appearance of both technical threats (resulting from design flaws or bugs) and non-technical threats (such as bribery).

Regarding the technical threats [31], a STRIDE (Spoofing, Tampering, Repudiation, Information disclosure, Denial of service and Elevation of privilege threats) sweep analysis on the two trust boundaries of an app ecosystem gives a wide range of threats, but only those reputation-related are stated below.

Thus, we can find the following attacks carried out during the publication of app description and reputation:

- Spoofing the application store interface, so users see false descriptions and reputations of apps.
- Tampering with the application store interface, changing descriptions and reputations.
- Preventing users from browsing app descriptions by overloading the application store

On the other hand, the following attacks are carried out during the acceptance of comments and complaints:

- Spoofing the application store, so the user submits comments/complaints in the wrong place
- Tampering with the application store interface, changing or removing complaints
- Submitting positive feedback information and denying this later
- Preventing users from submitting comments/complaints by overloading the application store with comments

Furthermore, a non-technical threat analysis [32] is harder to carry out, as it depends so much on attacker's roguery. Nevertheless, as reputation systems gather and process sensitive personal data, it is at least obvious that they must comply with the appropriate legislation and good IT security practice. It is for that reason that ENISA report on trust and reputation models [33] urged European Commission bodies responsible for legislation in relation to privacy (DG Justice), as well as national legal DPAs (Data Protection Authorities), to address the risks involved when disclosing sensitive information like privacy and trust. In this direction, the General Data Protection Regulation was released in 2012 (although the adoption is aimed in 2014 and the regulation is planned to take effect in 2016 after a transition period of 2 years).

4. Reputation Models

Reputation management systems have been successfully applied to several contexts and scenarios. Even though each of them has it owns peculiarities, and are adapted to face different kinds of challenges, they are aimed to compute recommendations, which are based on past interactions, in order to predict the behaviour of a given subject (an application, a service, a provider, etc.). In this section, we briefly introduce these systems and classify them according to the way of collecting feedbacks (i.e. users' recommendations) and we present different ways of aggregating those feedbacks to compute reputation values.

4.1. Feedback collection

Despite the reluctance of application platforms to provide technicality about their reputation systems, Farmer and Glass [34] came up with the five common reputation models used by current main providers of online products and services (not only application providers), based on the observance of the way feedback is collected from customers and how reputation scores are calculated and distributed. In the following, these five models are outlined.

4.1.1. Vote to promote

Users are allowed to vote for an item, and the number of votes is used as a ranking score. In some cases they are also allowed to vote against that item or even to retract their votes, so the score is decreased accordingly. Facebook [35] uses this system with its *Like* button, and YouTube goes further on using this information for ranking videos and giving the most popular ones a higher position in search results.

4.1.2. Content rating and ranking

Instead of simply letting users vote for or against items, this model allows them to rate items on a numeric scale, usually represented as stars, bars or numbered scales. Then, an average score is calculated from the collected votes to present the overall reputation. Wikipedia website [36], for instance, uses this system to allow the users to rate its articles.

4.1.3. Content reviewing and comments

Users are asked to rate different aspects of the item (i.e. quality, price, delay) and can even write a more detailed description of their viewpoint in a free form text field. The average score is calculated for every aspect and all the different written reviews are compiled and accessible by just a click. Frequently there is also an overall score that can be used for content filtering or ordering, as when Amazon [37] ranks its products.

4.1.4. Incentive points

This reputation system differs from the previous ones in the fact that users are not required to rate items. On the contrary, the system exploits the user's desire to achieve a better ranking, leading them to fulfil certain tasks. Some user actions are defined so as to be worth a number of points, and some others (i.e. lack of activity) reduce the total amount. The higher the number of points accumulated, the better the user reputation is. To mention some examples, LinkedIn [35] prompts users to provide information about themselves in order to achieve a 100% profile; and when playing some social games points can be redeemed to access services for free

4.1.5. Quality karma

Users are allowed to rate sellers only once the transaction is completed, so the posting of fake reviews does not compromise the system. Sellers are rated on criteria such as their number of transactions, for how long they have been sellers, the quality of the product, communication between buyer and seller, dispatch time or postage charges. On eBay [37], for instance, sellers with higher reputation score and more positive feedback are more likely to be chosen by users, as they are perceived as trustworthy traders.

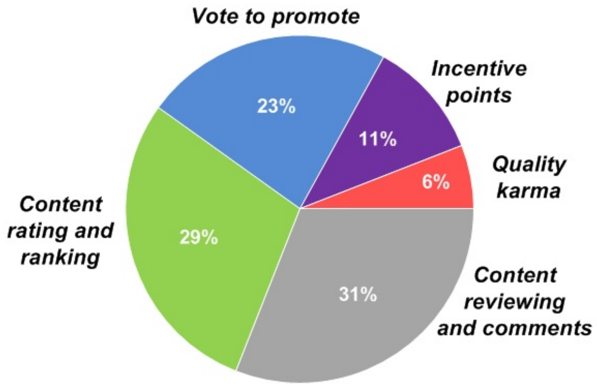


Figure 4: Use of reputation models by a range of well-known global websites

Coming back to the context of HBB domain, it is easy to see that any of the five reputation models previously described, or even any combination of them, could be applied to the applications offered to end users through HBB platforms. It would be up to the app ecosystem designers to decide the grade of complexity of the trust and reputation model applied. Figure 4 shows to what extent each reputation model is currently used, being “content reviewing and comments” the most popular one, followed closely by “content rating and ranking”.

4.2. Aggregate recommendations

There are many ways of aggregating the collected recommendations to compute the reputation value for a given item. The mechanisms to compute reputation not only differ on the way of performing the reputation calculation but also in the resources they need to work. Moreover, in addition to the recommendations they may need other contextual information, such as users’ preferences, or the identifier of the recommenders.

As presented in [38], there is not a perfect reputation computation engine suitable for all conditions. Some of them tend to be simple to require less resources to work, whereas other are designed to avoid malicious users, or they may only provide good results if there are several users using the system. In the following we summarize four mechanisms analyzed in [38] used to aggregate recommendations.

4.2.1. Average

This mechanism calculates the arithmetic mean of all the available user recommendations. This is the most straightforward way to compute reputation from recommendations, since it does not require taking into account other values but the recommendations themselves. Even though this mechanism is the best choice for some scenarios due to its simplicity and scalability, it is not resilient to malicious users which try to increase or decrease the reputation of a given item by providing biased recommendations [14].

4.2.2. Weighted Average

To avoid malicious recommendations, this mechanism extends the previous one by assigning a weight to each recommender. Those weights are updated according to the accuracy of the recommendations given by each recommender. The accuracy of a recommendation is, in turn, calculated based on the deviation of that recommendation with regard to the rest of recommendations (e.g. a recommendation that is far apart from the rest is likely to be biased). The reputation values are hence computed performing a weighted average. In this case, it has been proved that biased recommendations could be avoided (to some extent), although the complexity of the model increases.

4.2.3. Preferences Weighted Average

The previous mechanism does not take into consideration that each user could have a different expectation about the same service, and therefore providing and expecting different recommendations. For example, a user could rate a fairly good but expensive application as bad, not because she is a malicious recommender, but because she gives more relevance to other aspects such as its price, amongst others.

The preferences weighted average mechanism introduces an additional parameter in the computation to measure the similarity of each pair of users. This similarity is based on pre-defined preferences of the users [39]. For instance, they may select a set of parameters according to their predilection when they register into the system. Even though this mechanism requires further computation, it provides customized reputation values, enhancing this way the user experience.

4.2.4. Users Weighted Average

The previous mechanism calculates a customized reputation value assuming that all users having similar preferences have the same expectation and give similar recommendations. However, there could be a huge variety even within the same range of preferences, or there could be users that are not willing to provide their preferences, either for privacy concerns, or because they see this as an irrelevant aspect.

The users weighted average mechanism goes one step further to compute the similarity for each pair of users [40]. The similarity between a pair of users is based on the likeness of the recommendations already given by these users. For example, two users would have high level of similarity if they like and dislike the same applications.

5. Privacy-preserving reputation management

The generic steps to be carried out by reputation systems can be roughly described as follows. Firstly, a rating process enables users to provide feedback on their experience while interacting with products, services, and providers. Then, a reputation function aggregates the feedback values and calculates reputation scores. Finally, a query process allows other users to check the reputation of an item and evaluate its trustworthiness.

As described in Section 4.1, there are many ways of aggregating the collected recommendations to compute reputation values. Some of the described methods are able to provide enhanced and customized reputation values, although they require additional information of the users and/or recommenders.

In the HBB context, these improvements would require the application store to know information details of the users (their identifier, the recommendations they have provided, their preferences, similarity between the users, etc.), which directly results on a of violation of users' privacy. Moreover, some research works [17] point that users may avoid providing honest recommendations about a service in fear of retaliation if recommendations cannot be provided in a privacy-preserving way.

In this section we present a privacy-preserving reputation framework to be integrated within the HBB context. This reputation framework is able to compute personalized reputation values, based on the similarity between users' recommendations (i.e. using the *users weighted average* mechanism), but without revealing the real identity of those users to the application store.

5.1. Adding a trusted party

One of the main goals of any privacy-preserving system is to hide the real identity of the users when interacting with online services [41]. In this scenario, the main concern is to hide the users' real identifier to the application store in such a way that these users can interact with the services in a privacy-preserving way. Nevertheless, the application store has to enforce authentication mechanisms in order to guarantee that users do not misuse their services.

In scenarios where the application store is in charge of both the authentication and the reputation management, as shown in Figure 5a, there is not an easy way for the users to avoid being traced, since their interactions with the application store are linked to their accounts, which are managed by the application store, indeed.

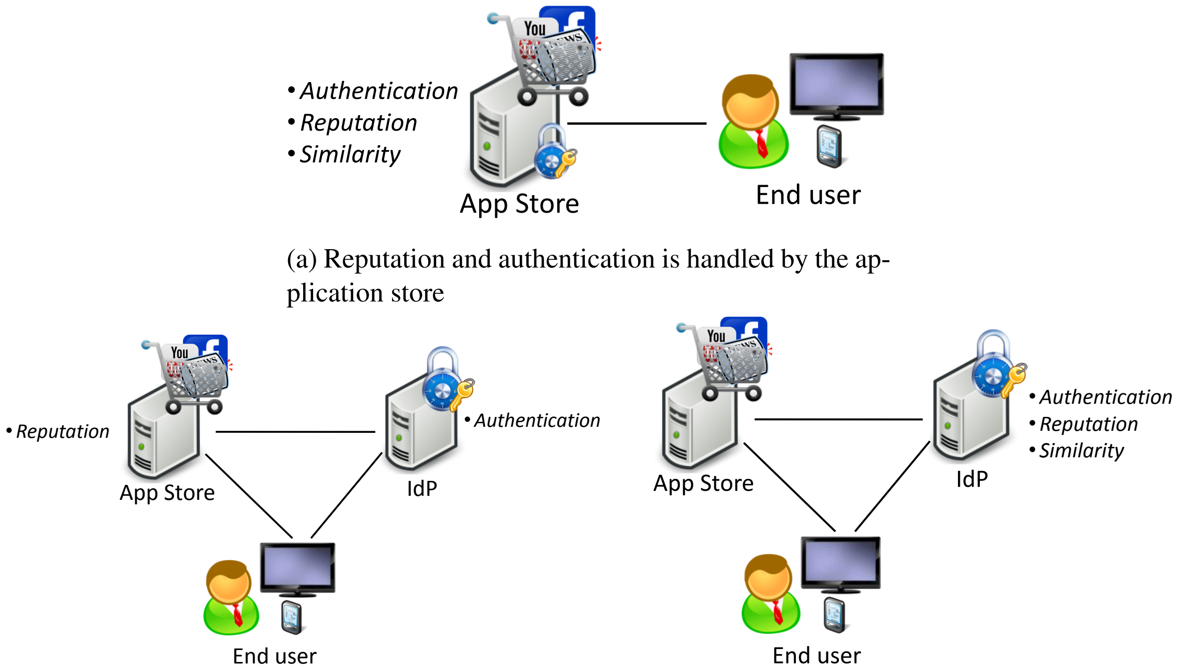
To avoid that, in the proposed framework we introduce an external trusted party (i.e., not colluding with the application store) known as identity providers aimed to manage the identity of the users and to provide authentication capabilities [42]. Assuming the existence of entities or services working collaboratively without colluding with each other has been widely accepted by the research community. For instance, within the identity management field [43], scenarios where authentication is delegated to an external trusted party are becoming increasingly common.

Additionally, this assumption is widely extended in many cryptography works, where authors analyze their work against semi-honest (or honest-but-curious) adversaries. For example, in multi-party computation [44], different entities collaborate with each other to achieve a common goal without revealing each other more information than strictly required to reach such goal, and many work in this regards relax the security model properties, assuming that the different entities will not collude [45, 46, 47].

In our case, for those users who want to access a restricted service of the application store (e.g. purchasing or downloading an application), the authentication process is delegated to their identity provider. The identity provider then authenticates the users and provides a token with a pseudonymous identifier to the application store.

Thus, the application store can verify that an authentication process has been performed whereas the users can access the services in a private fashion, since the application store does not know the details of the users' accounts (being the pseudo-identifiers the only information it receives from the users). For example, they could provide recommendations about an application to the application store without compromising their privacy. Furthermore, if a different pseudonym is provided for each user interaction, the application store cannot relate such interactions between each other.

This mechanism preserves the privacy of the users' identifiers, but on the other hand the application store cannot compute customized reputation values, as depicted in Figure 5b. Due to the fact that the application store cannot trace which user has provided each recommendation, it cannot relate the users according to the provided values, determining their similarity.



(a) Reputation and authentication is handled by the application store

(b) Authentication is delegated to an identity provider

(c) Reputation and authentication is delegated to an identity provider

Figure 5: Reputation management scenarios for an application store adding an identity provider (IdP)

A straightforward solution would consist of delegating the reputation management functionality to the identity provider [48, 49], as shown in Figure 5c. The users would send their recommendations to their identity provider, and this one would aggregate them. The identity providers would even communicate with each other to share recommendations in a privacy-preserving way to increment the amount of recommendation sources [38].

However, even though the identity providers are considered trustworthy parties, many users would still be reluctant to send their recommendations to the identity providers due to the lack of privacy that this mechanism presents. In the end, using this mechanism, the identity providers know the real identifier of the users and the recommendations provided by each of them.

In general, private information about the users (e.g. account identifier), and their interactions (e.g. which reputation values they provide) should not be known by the same entity. To preserve the privacy of the users in this scenario, we consider that the following privacy-related properties must be fulfilled in the reputation framework.

- **P.1** The application store must not know the real identity of the users. Pseudonyms should be used to hide the real users' identifiers.
- **P.2** The application store must not know whether two different interactions (e.g. a user supplies recommendations of two applications) have been performed or not by the same user.
- **P.3** The identity provider must not know the recommendations given by each user.
- **P.4** The identity provider must not be able to infer how the users are related to each other. In other words, the identity provider cannot know the similarity between two users.
- **P.5** The application store could know the recommendations provided by the users since they are provided in an anonymous way.

	Identity provider	Application store
real users' identity	✓	✗
similarity between users	✗	✓
users' recommendations	✗	✓
relate two users' interactions	✓	✗

Table 1: Privacy-preserving properties of both the identity provider and the application store

P.6 The application store could know the similarity between two users given its pseudonymized identifiers to provide customized reputation values.

P.7 The identity provider could know the real identity of the user as well as other authentication information.

P.8 The identity provider can correlate different users' interactions.

As shown in Table 1, if these properties are fulfilled in the reputation framework, neither the identity provider nor the application store have enough information about the users and their interactions to compromise their privacy. None of them could determine which recommendations a user has provided given his or her real identifier. Moreover, none of them could discover how the users are related to each other. How the proposed framework enforces the fulfillment of these properties, even in presence of adversaries, is analyzed in Section 6.

5.2. Proposed framework

In order to compute customized reputation values but maintaining the privacy properties previously described, the identity provider and the application store have to collaborate with each other but without revealing any private information. Hence, either the identity provider or the application store should be able to perform operations that require as input both their own data and the data which must be only known by the other part.

That could be achieved using advanced encryption techniques, such as homomorphic encryption [19]. Homomorphic encryption is a cryptographic technique which allows performing certain operations over encrypted data, obtaining an also encrypted result which matches the results of operations on the plain data. For example, it could add two encrypted values without decrypting them, obtaining as a result the encryption of the addition of those plain values.

Using this technique, the application store would encrypt the feedbacks provided by the user, and send them to the identity provider in such a way that the latter could compute the similarity between the users, without knowing either those feedbacks or the result of the operation (Figure 6). In the following we describe the processes in more detail.

5.2.1. Authentication

The application store delegates the authentication process to the identity provider. The identity provider, as in any common identity management scenario, hides the users' identity by using different pseudonyms when they access to different services [50].

In order to relate the different interactions of the users, the identity provider maintains the list of pseudonymous used by each user. In this way, it could determine whether two different interactions have been done or not by the same user, without knowing details of such interactions. For example, it could know which pseudonymous a user has used to provide each of these feedbacks, without knowing the feedback submitted. The authentication process is as follows.

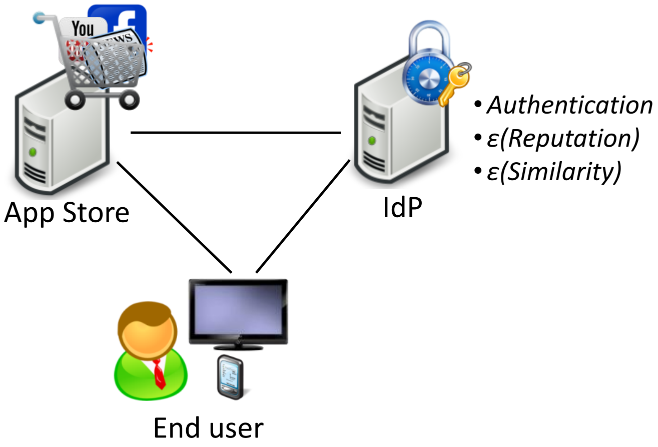


Figure 6: The identity provider handles encrypted reputation information

a.1 A user tries to access the application store through her HbbTV device (e.g. her TV).

a.2 The application store redirects the user to her identity provider along with an authentication request message as defined by any standard protocol, such as SAML [51] or OpenID [52].

a.3 The identity provider authenticates the user, for example validating her username/password.

a.4 The identity provider generates a pseudo-random identifier. This identifier is stored and linked with the user identity, in order to relate the different users interactions later on.

a.5 The identity provider redirects the user back to the application store, including an authentication statement as defined by any of the aforementioned standards. The authentication statement includes the generated pseudo-random identifier.

a.6 Once the application store receives and validates the authentication statement it responses the user accesses request, showing the application store website.

5.2.2. Obtain an application and its reputation

Before downloading any application, the users usually want to know more information about it, besides the description specified by the vendor. While they browse the application store, they expect ratings which summarize the feedbacks given by other users to those applications.

As we commented, to enhance the accuracy of these ratings, they should be computed according to the similarity between users, but taking into account privacy issues. The similarity value between each pair of users is maintained and updated by the identity provider using homomorphic techniques, which means that it could only be decrypted and known by the application store, as we show in next subsection. In this way, the application store could request similarity between users given their pseudonyms to the identity provider, even though the identity provider cannot decrypt the similarity values. In the following, the process of collecting and computing the reputation information is described.

b.1 The authenticated user is browsing the application store website and wants to know the reputation of an application (or set of them).

b.2 The application store retrieves the list of feedbacks about the application, and the pseudo-identifier of the users who provided such feedbacks.

b.3 The application store sends a request to the identity provider asking for the similarity between the authenticated user and each of the users who provided a feedback for the requested application.

b.4 The identity provider obtains the real users' identifier from the pseudo-identifiers received.

b.5 The identity provider recovers the encrypted similarity value between the authenticated user and each of the specified users, and provides these values to the application store.

b.6 The application store decrypts the similarity values and computes the personalized reputation accordingly as described in equation 1, where $Reputation_{user_j}$ is the reputation value customized for user j, $similarity(user_i, user_j)$ represents the similarity value between users i and j, and Rec_{user_i} is the recommendation provided by user i.

$$Reputation_{user_j} = \frac{\sum_{i=1}^{n} similarity(user_i, user_j) \cdot Rec_{user_i}}{\sum_{i=1}^{n} similarity(user_i, user_j)} \quad (1)$$

b.7 The application store provides the reputation of the application to the user.

5.2.3. Provide feedback

Since the similarity between two users is based on the users' feedbacks, the similarity information has to be updated when a new feedback is received. The similarity information is maintained by the identity provider, but only the application store must know the feedbacks provided by the users.

To preserve the privacy of the feedbacks, the application store encrypts them before sending them to the identity provider. Even though the identity provider cannot know the feedback values, it knows who has provided each of them, since it could relate the pseudonyms with the users' identity. Using homomorphic encryption techniques, the identity provider could update the similarity information according to the feedbacks, but without decrypting them. In the following, the process of providing feedbacks is described:

c.1 After playing around with the application, the user (already authenticated) wants to provide her feedback about the installed application. Then, using the application store website she sends her rating.

c.2 The application store saves the received feedback to be used for computing future reputation values. Additionally, it encrypts such feedback.

c.3 The application store sends the encrypted feedback to the identity provider along with the identifier of the application and the pseudo-identifier of the user.

c.4 The identity provider recovers the identifier of those users who provided a feedback for such application, along with their encrypted feedbacks. From the encrypted feedbacks, it computes the similarity as shown in Section 5.3. Additionally, the identity provider stores the encrypted feedback received, which will be used for computing similarity for other users in the future.

Figure 7 depicts the described processed.

5.3. Private similarity computation

Homomorphic encryption is a cryptographic technique that enables mathematical operations to be performed over encrypted data. For instance, it could take two encrypted values as input and perform the addition operation without decrypting them, obtaining the result also encrypted.

There is a lot of research work going towards improving this technique [53, 54, 55] due to its extensive applicability in many different fields. However, currently homomorphic encryption has only be proved to be efficient and practical for performing additions and multiplications over

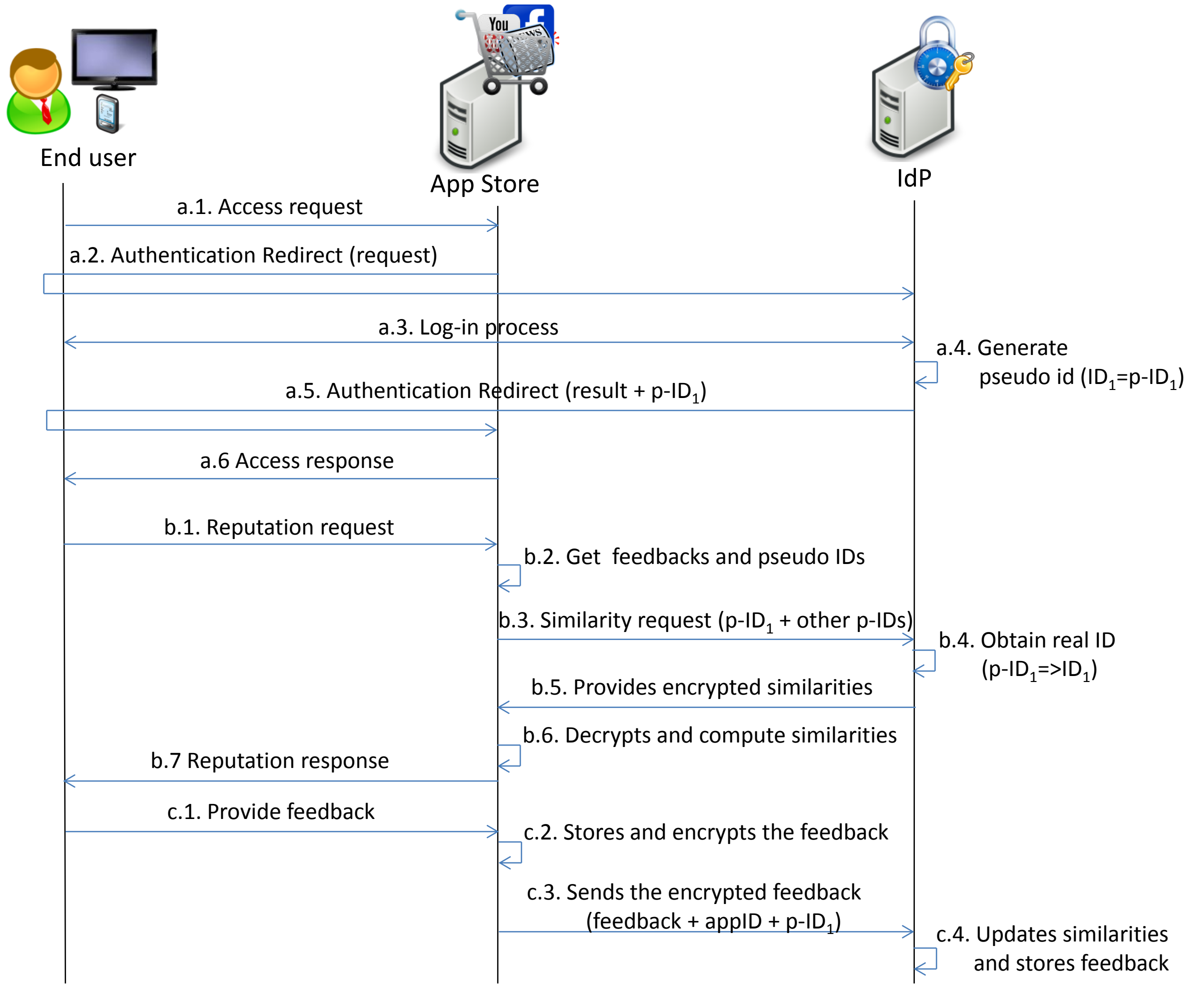

End user
App Store
IdP
a.1. Access request
a.2. Authentication Redirect (request)
a.3. Log-in process
a.4. Generate pseudo id (ID1=p-ID1)
a.5. Authentication Redirect (result + p-ID1)
a.6 Access response
b.1. Reputation request
b.2. Get feedbacks and pseudo IDs
b.3. Similarity request (p-ID1 + other p-IDs)
b.4. Obtain real ID (p-ID1=>ID1)
b.5. Provides encrypted similarities
b.6. Decrypts and compute similarities
b.7 Reputation response
c.1. Provide feedback
c.2. Stores and encrypts the feedback
c.3. Sends the encrypted feedback (feedback + appID + p-ID1)
c.4. Updates similarities and stores feedback

Figure 7: Privacy-preserving reputation framework processes overview

bits [47, 56]. That is, current mechanisms can only efficiently operate with encrypted bit values (0 or 1).

In this section, we present a mechanism to privately compute the similarity between users using homomorphic encryption schemas based on operations over bits. Making use of fully homomorphic encryption schemas, we are able to perform implicit plaintext additions and multiplications while manipulating only ciphertexts. Summarizing, a fully homomorphic encryption mechanism allows the following operations over bits (being b_1 and b_2 two bits values, whereas $E_k(b_i)$ represent the encryption of the bit b_i using the cryptographic key k).

- Additive operation (XOR equivalent).

$$E_k(b_1) \oplus E_k(b_2) = E_k((b_1 + b_2)\ mod\ 2)$$

- Multiplicative operation (AND equivalent).

$$E_k(b_1) \otimes E_k(b_2) = E_k((b_1 \cdot b_2)\ mod\ 2)$$

Nevertheless, expressing the similarity between two users as a bit would decrease the accuracy of the reputation system, since it could only indicate extremes, such as either they are fully similar (*similarity* = 1) or not at all (*similarity* = 0). Instead, we envision the similarity as a natural number within a given range $[0, max\_similarity\_value]$, being *max_similarity_value* (MSV) the maximum value that the similarity can take. Additionally, we choose MSV such that it fulfils $\exists\, n \in \mathbb{N} : 2^n - 1 = MSV$. In this case, the higher the similarity number is, the more similar a pair of users are.

To apply the aforementioned homomorphic encryption operations, the similarity is represented as a bit array whose length is $log_2(MSV + 1)$, according to its binary representation. For example, being $MSV = 15$ the representation of a similarity value of 9 would be [1,0,0,1]. Each bit of the array is encrypted independently. For instance, a bit array of length L is encrypted as $[E_k(b_1), E_k(b_2), \ldots, E_k(b_L)]$.

There is an encrypted similarity bit array for each pair of users, which are maintained by the identity provider. These arrays are created when a user is registered in the system (for example, when she creates a new account in the identity provider), taking a default value. The default value could be for example the mean value between 0 and MSV, or it could be enforced by the application store if it provides a default encrypted similarity bit array. It is worth mentioning that if these arrays are created by the identity provider they would not be encrypted, although they would be encrypted as soon as an update on the similarity value has to be performed, as shown below.

The similarity between two users has to be updated according to the closeness between their feedbacks. Therefore, these feedbacks have to be compared by the identity provider even though they have been encrypted by the application store before sending them. To than end, the feedbacks are represented using a bit array of length *possible_feedbacks_values* (PFV), but instead of representing the binary representation of the feedback value, each position of the array represents whether this value is the actual feedback. That is, the array can only have a bit set to '1', where the position of this bit indicates the value within the range. For example, if the system allows to send a rating from 0 to 5 ($PFV = 6$), the representation of a score of 4 would be [0,0,0,0,1,0], whereas a score of 2 would be [0,0,1,0,0,0].

Since the values are encrypted, the homomorphic encryption operations are used to compare two feedbacks using the algorithm 1.

```
Input: $feedback_1 := encryptedBitArray[1..PFV]$,
       $feedback_2 := encryptedBitArray[1..PFV]$
$feedbacksComparison \leftarrow 1$;
for $i \leftarrow 1$ to $PFV$ do
    $feedbacksBitComparison \leftarrow feedback_1[i] \oplus feedback_2[i]$;
    $feedbacksComparison \leftarrow feedbacksBitComparison \otimes feedbacksComparison$;
end
return $feedbacksComparison$
```

Algorithm 1: Compare two private feedbacks

This algorithm returns an encrypted '1' ($E_k(1)$) if the two feedbacks are equal and an encrypted '0' ($E_k(0)$) otherwise. This value cannot be decrypted by the identity provider to preserve the privacy of the feedbacks, so it cannot know whether it should increase or decrease the similarity. However, this value could be inserted as input to the algorithm 2 to update the similarity accordingly.

```
Input: $feedbacksComparison \in \{E_k(0), E_k(1)\}$,
       $similarity := encryptedBitArray[1..MSV]$
$and\_value \leftarrow similarity[1] \oplus 1$;
$and\_value \leftarrow and\_value \oplus feedbacksComparison$;
for $i \leftarrow MSV$ to 1 do
    $similarity[i] \leftarrow and\_value \oplus similarity[i]$;
    $xor\_value \leftarrow similarity[i] \oplus 1$;
    $and\_value \leftarrow and\_value \otimes xor\_value$;
end
```

Algorithm 2: Increase similarity according to feedbacks comparison

At the end, the encrypted similarity value remains equal if the input value is an encrypted '0' ($E_k(0)$), but it is incremented by 1 if the input is an encrypted '1' ($E_k(1)$). Since these operations are performed over encrypted data, the identity provider cannot know whether it has just increased the similarity values or not, hence preserving both the feedbacks values provided by the user and their resulting similarity. It is also worth mentioning that adding the first operation of the algorithm ($and\_value \leftarrow 1 \oplus similarity[1]$) we prevent the similarity value to increase if it has reached its maximum value, producing inconsistent results.

Additionally, the similarity values should be decreased to punish the similarity of those users who do not provide similar feedbacks. An special encrypted bit, entitled *isZero*, is created to determine whether the similarity has already reached a value of 0 and could not be further decreased, also to prevent inconsistent results. Algorithm 3 explains how to decrease the similarity value if the feedbacks provided do not match and update the *isZero* bit value in a privacy-preserving way.

```
Input: $feedbacksComparison \in \{E_k(0), E_k(1)\}$,
       $similarity := encryptedBitArray[1..MSV]$, $isZero \in \{E_k(0), E_k(1), 0\}$
$and\_value \leftarrow isZero \oplus 1$;
$feedbacksComparison \leftarrow feedbacksComparison \oplus 1$;
$and\_value \leftarrow and\_value \otimes feedbacksComparison$;
for $i \leftarrow MSV$ to 1 do
    $similarity[i] \leftarrow and\_value \oplus similarity[i]$;
    $xor\_value \leftarrow similarity[i] \oplus 1$;
    $and\_value \leftarrow and\_value \otimes xor\_value$;
end
$isZero \leftarrow similarity[1] \oplus 1$;
for $i \leftarrow 2$ to $MSV$ do
    $isZero \leftarrow isZero \otimes similarity[i]$;
end
```

Algorithm 3: Decrease similarity according to feedbacks comparison

This algorithm decreases the encrypted similarity value if the input value is an encrypted '0' ($E_k(0)$), and remains equal if the input is an encrypted '1' ($E_k(1)$). However, the reputation framework might not want the similarity to be so drastically reduced. For example, the system could prefer reducing the similarity every once in a while to prevent the users to lose their similarity very often. To determine when to decrease the similarity between users we propose the following three mechanisms.

- The *decrease similarity algorithm* is launched just after the increase algorithm, when a feedback is received. In this sense, when the similarity is being updated it decreases by

1 if the feedbacks do not match and increases by 1 if they match. This requires higher computational resources, and the similarity would likely to have a value of zero for most of the users pairs.

- The *decrease similarity algorithm* is never launched. In this sense, the similarity would start with 0 as default, and it would increment as long as two feedbacks match, but no punishment to the similarity would be applied besides not incrementing if the feedbacks do not match. It increases the performance of the reputation framework, but it poorly adapts to the changes in the users' interests. For example if a pair of users have high similarity in the past but they do not share interest any more.

- From time to time the identity provider launches the *decrease similarity algorithm* with a fixed input of '1' to decrease the similarity for all the users pairs. In this way, there is not punishment if the feedbacks do not match, but after some time the similarity values are decreased. The similarity of those users that no longer provide similar feedbacks would be decreased whereas it increases for those that are constantly providing similar feedbacks.

6. Security Analysis

This section presents a security model in order to analyze the behavior and resilience of the proposed framework against malicious users or entities. Adversaries considered in this analysis aim to compromise the privacy of the users, willing to find out private information that they are not supposed to know.

Besides, reputation threats are analyzed in order to describe how the system behaves against malicious users, trying to distort the reputation of applications, hence decreasing the utility of the reputation framework. These malicious users attempt to submit biased recommendations, maybe massively, in order to increase or decrease the reputation values.

6.1. Security model analysis

To preserve the privacy of the users, a set of properties that must be fulfilled by the system was introduced in Section 5.1. How these properties are materialized in the proposed framework, even in the presence of adversaries is analyzed in the following. Here, the adversary would compromise either the application store or the identity provider willing to find out private information that these entities are not supposed or allowed to know. This includes private information they can obtain directly, or inferring it from other information they may have access to.

6.1.1. Security model assumptions

In this security analysis, adversaries' efforts take the form of attempting to discover secret data, by inferring data from other information they have access to. We therefore focus on honest-but-curious adversaries. In the honest-but-curious model [44] all parties are assumed to follow protocols properly, yet they keep all inputs and intermediate results willing to secretly infer some knowledge they are not supposed to know.

Adversaries aimed to corrupt some of the data in the system, spoofing the identity of a message sender or receiver, or making resources unavailable are out of the scope of this security analysis. Hence, we consider that our adversaries cannot monitor, alter or drop messages exchanged through the communication channels.

We assume that a realistic adversary in this context is polinomially bounded. That is, our adversary cannot consider all possible inputs to break the encryption algorithm. Furthermore, our adversary does not have access to the private keys of other entities. That means, private keys are only known by the entity that the keys belong to. Furthermore, the messages are encrypted in such a way that they cannot be compared to each other to infer the contained information.

Nevertheless, an adversary has access to any piece of information that the compromised entity would have access to, although we assume that an adversary cannot compromise more than one entity. In other words, we assume that entities cannot collude.

We summarize how the adversary model behaves in our environment in the following way:

- An adversary can compromise the application store or the identity provider, but not both. Therefore, application store and identity provider do not collude.
- The adversary is considered honest-but-curious.
- The adversary is willing to break the privacy-related properties presented in section 5.1. In other words, the adversary would try to obtain more information than the compromised entities are supposed to know.
- The adversary only knows the private key of the compromised entity. That is, the adversary cannot decrypt a message that has been encrypted by other entity.
- The adversary cannot monitor or control the communication channel.
- The adversary can only execute polynomially bounded algorithms.
- The adversary would have access to any piece of information that the compromised entity has access to.
- From the comparison between encrypted messages, the contained information cannot be inferred.

6.1.2. Application store tries to compromise users privacy

According to the defined privacy-related properties, the application store knows the recommendations provided by any pseudo identifier (P.5), as well as the similarity between pseudo identifiers (P.6). In order to properly validate the proposed framework, we have to analyze the case where a corrupted application store is willing to get further private information.

Let us assume that an adversary is controlling the application store. In this case, we need to analyze that the application store does not learn anything but the information determined by the privacy-related properties. Hence, it should not be able to discover neither the real users identifier (P.1), nor the relations between users' interactions (P.2), otherwise it could trace the different recommendations provided by each user.

The real identity of the users is not released to the application store at any time. As we describe in steps a.2 to a.5, the users are authenticated using their identity provider, and this identity provider does not release more private information than the generated pseudonyms. Hence, as long as the pseudonyms are not generated from the real identifiers using an algorithm that can be reverted in polynomial time, the adversary cannot known the real identifier of a user (P.1).

However, we still have to analyze whether the adversary can correlate users' interactions by inferring the information retrieved from the identity provider. The only part where the application store gets users' information from the identity provider is in step b.3. There, the application store can retrieve the similarity between pseudo identifiers from the identity provider. On the other hand, since the users employ pseudonyms, the application store cannot directly determine whether different pseudo identifiers belong to different users or to the same one. Yet, we need to study whether the adversary can infer whether two pseudo identifiers belong to the same user from the similarity information (P.2).

The adversary is able to indirectly compare the similarity values of two users from a third one. That is, if the similarity between pseudo-identifiers A and B is equal to the similarity between pseudo-identifiers A and C, there is certain likelihood that the pseudo identifiers B and C belong to the same user. However, if the number of users is much greater than the possible similarity values (which is the case in common reputation scenarios), the resultant probability is negligible.

Furthermore, the identity provider could easily deny random requests from the application store. In fact, in order to follow the process correctly, the application store only needs the similarity between the pseudonym of a recently authenticated user and others. Hence, the identity provider could deny similarity requests not asking for a pseudo-identifier recently generated (i.e. belonging to a user that has been recently authenticated).

6.1.3. Identity provider tries to compromise users privacy

Even though the identity provider is supposed to be a trustworthy entity, in the sense that it is not releasing private information to others, it might want to obtain as much users' information as possible to be used in its own benefit (for example for marketing or advertisement purposes).

According to the defined privacy-related properties, the identity provider knows the real identity of the users (P.7), and can correlate different users' interactions (P.8). But let us assume that the identity provider is compromised, and therefore an adversary is controlling it. In this case we need to investigate whether the adversary is able to know either the recommendations the users have provided (P.3), or the similarity between them and, hence, between users (P.4).

On the one hand, since the recommendations are directly provided to the application store by the users, the adversary does not have access to them. On the other hand, the adversary has access to the encrypted recommendations, once they are released by the application store to the identity provider in step c.4. Nonetheless, as the adversary does not know the application store's private key, the identity provider cannot reveal the encrypted recommendations values (P.3).

Moreover, even though the adversary could fully control the process of computing the similarity between users, she cannot determine the actual similarity values (P.4). The similarity computation is directly done over encrypted data by using homomorphic encryption techniques. Hence, the adversary does not have access to the decrypted similarity values at any moment.

Besides, according to our design, the similarity values are encrypted in such a way that the encrypted values the adversary has are completely meaningless without having the application store's private key. Furthermore, as we have stated in the assumptions, encrypted similarities values cannot be compared to each other to infer the contained information.

6.2. Reputation threats analysis

Next, we will analyze how the framework behaves against reputation threats where malicious users aim to ill-intentionally distort the reputation values of the applications offered by the application store. These malicious users attempt to submit biased recommendations in order to influence in the reputation values, hence decreasing the utility of the reputation framework.

6.2.1. Attacker distorts reputation by providing highly biased recommendations

In this case, the goal of the attacker is to reduce the usefulness and relevance of the results produced by the reputation framework by unfairly increasing or decreasing the reputation score of certain applications. For example, the author of certain application might want to increase the reputation of the applications she has developed by providing fake recommendations. That is, acting as a user of those applications, hiding behind the anonymity the framework provides.

In step c.1, the user sends her recommendations to the application store. We need to therefore analyze whether the user can send unlimited number of recommendations in such step.

When a user requests access to the application store (step a.1) she needs to be authenticated against the identity provider (steps a.2 to a.5). In other words, even if the user is anonymous to the application store (P.1), as it only knows the generated pseudonym, the user needs an account in the identity provider to supply recommendations. Thus, the identity provider, since it knows the identity of the user (P.7), can limit the number of recommendations a user can supply about an application.

On the other hand, the malicious user would be able to create multiple fake accounts in order to feed the system with many biased recommendations by acting as many different users. Even if the framework does not prevent that, it is impracticable or useless in systems where many users are interacting. The impact of the recommendations that a single user could supply by manually creating accounts is in most cases negligible, as the malicious user has to compete with the recommendations of the rest of the users.

Moreover, the reputation of the HBB applications in the proposed framework is computed based on the similarity between users, as shown in step b.6. The profile of the accounts created only to provide biased recommendations would be unlikely similar to the profile of other regular users. That decreases even more the impact of the biased recommendations provided by any malicious user.

6.2.2. Attacker submits biased recommendations massively

As an extension of the previously described case, a malicious user, in order to be more effective on her purpose, would try to perform a Sybil attack [57]. In this scenario, a Sybil attack would consist on automatically creating a large number of accounts and using them to gain a disproportionately large influence on the reputation of some applications.

We need to study whether the proposed framework is able to prevent this attack by restricting the number of accounts a single user can create or interactions she can perform.

The identity provider could easily avoid massive account creation by deploying CAPTCHAs [58, 59] in the account creation process (which leaves out of the scope of this document). CAPTCHAs (Completely Automated Public Turing test to tell Computers and Humans Apart) avoid bots or scripts performing automatized processes by testing the users with a challenge that usually only humans can pass. This makes creation of multiple accounts difficult unless manually, whose consequences were discussed above.

In the same way, the application store could prevent automated interactions by also using CAPTCHAs when the users provide recommendations, in step c.1. This is useful in case the application store cannot guarantee that the identity providers are enforcing any mechanism to prevent massive account creation.

PFV	Toy	Small	Medium	Large
1	0.004 s	0.024 s	0.160 s	0.988 s
2	0.008 s	0.048 s	0.312 s	1.964 s
3	0.010 s	0.068 s	0.472 s	2.956 s
4	0.012 s	0.096 s	0.636 s	3.944 s
5	0.016 s	0.116 s	0.792 s	4.932 s

Table 2: Computation times of comparing two feedbacks

7. Experiments

This section describes the conducted testbeds aimed to demonstrate the viability of the proposed framework in terms of performance. The accuracy of the reputation model *Users Weighted Average*, which is the one used in this solution, has been already analyzed in [38, 60]. Therefore, we focus the experiments on analyzing the computation of the similarity between users in a private way, achieved using the mechanisms described in Section 5.3.

We make use of the implementation of the DGHV (Dijk, Gentry, Halevi and Vaikuntanathan) [61] scheme with fully homomorphic encryption capabilities described in [62]. That work provides an implementation of the key generation, encryption, decryption, addition, multiplication and ciphertext refresh procedures. It allows performing unlimited addition and multiplication on ciphertexts by reducing the amount of noise generated after performing such operations. The implementation is done with the SAGE 4.7.2 mathematical library under Python, and it is available in [63].

Making use of the operations allowed by that implementation, we have developed the algorithms described in Section 5.3. In the end, those algorithms consist of a set of additive and multiplicative operations, but performed over encrypted bits. In this sense, we are simulating the behaviour of the reputation framework when i) comparing two feedbacks and ii) when a similarity value is updated accordingly, so the elapsed times for each of these operations could be analyzed.

The tests have been performed in an Intel Quad-Core i5-2400 64 bits Processor at 3.10GHz with 8GB of RAM. We have performed tests with the different public key sizes tested in [62] (i.e. toy = 77KB, small = 437KB, medium = 2207KB and large = 10.3MB) so the results could be complemented with that work. Additionally, for each key size, we have run the tests setting different possible feedback values (PFV) and different maximum similarity value (MSV).

Table 2 summarizes the computation times resulted from comparing two feedbacks (algorithm 1). As it is shown, for example using 5 possible feedbacks values (e.g. rating the apps from 1 to 5 stars), and using a medium-size key for encryption, it takes 0.792 seconds to compare two encrypted feedbacks.

Table 3 summarizes the computation times resulted from increasing the similarity between two users according to the feedback comparison result (algorithm 2). For instance, having a MSV of 15 (i.e. the similarity values ranged between 0 and 15) and a medium-size key, the algorithm increasing the similarity takes 0.752 seconds to complete.

Finally, Table 4 summarizes the computation times resulted from decreasing the similarity between two users according to the feedback comparison result (algorithm 3). Following with the same example as before, having a MSV of 15 and a medium-size key, the algorithm decreasing the similarity takes 1.202 seconds.

MSV	Toy	Small	Medium	Large
1 (1 bit)	0.004 s	0.048 s	0.302 s	1.976 s
3 (2 bits)	0.008 s	0.068 s	0.452 s	2.904 s
7 (3 bits)	0.010 s	0.092 s	0.606 s	3.880 s
15 (4 bits)	0.014 s	0.112 s	0.752 s	4.908 s
31 (5 bits)	0.016 s	0.136 s	0.904 s	5.924 s

Table 3: Computation times of increasing similarity between two users

MSV	Toy	Small	Medium	Large
1 (1 bit)	0.008 s	0.044 s	0.304 s	1.948 s
3 (2 bits)	0.012 s	0.088 s	0.604 s	3.876 s
7 (3 bits)	0.016 s	0.136 s	0.904 s	5.844 s
15 (4 bits)	0.024 s	0.180 s	1.202 s	7.824 s
31 (5 bits)	0.032 s	0.228 s	1.506 s	9.704 s

Table 4: Computation times of decreasing similarity between two users

It is worth mentioning that these algorithms are executed by the identity provider when a feedback is received, not when a reputation value needs to be computed. In other words, the algorithms are executed offline and not when a user is waiting for a reputation response. Thus, taking a medium-size key, the algorithms could be completed in a reasonable time frame whereas the users do not appreciate further delay on their daily interactions with the application store.

After having the framework running for a while, and the number of users growing, obviously more and more computation is required to manage customized reputation values. For example, obtaining the reputation of an application which has been rated by several thousands of users would result on having impracticable computation loads if large encryption keys are used. Likewise, this may happen when having to update the similarity between those users upon reception of a feedback.

Nevertheless, several mechanisms can be applied to avoid such overload in long-scale systems. One approach may consist of taking a small subset of recommendations to compute the reputation of a highly rated application (e.g. the most recent ones). This mechanism, known as selective aggregation [64], would highly decrease the overload whereas the accuracy on the reputation values would not be deeply affected, as usually a small portion of the recommendations is representative enough.

8. Related work

In this section we present some research works related to the fields of identity management, trust and reputation management, homomorphic encryption and app ecosystems. As we will see, none of them gathers the key features of the approach described in the paper at hand, namely: providing user-tailored reputation scores based on similarities between users, while preserving the privacy of those end users.

Thus for instance, authors of [65] present a trust and reputation model making use of homomorphic encryption techniques in order to aggregate recommendations from end users about

Care Service Providers (CSP), in a privacy-preserving way. Yet, this solution offers an homogeneous set of reputation scores regarding CSPs to every user. In other words, it is unable to provide customized reputation values to each individual user.

On the other hand, [38] describes a set of mechanisms in order to compute user-tailored reputation scores, in the context of OpenID, by aggregating recommendations or feedbacks from end users, based on either the preferences or the similarities amongst such users. Moreover, it proves that this kind of personalized reputation scores entails a higher satisfaction for the end users. Albeit, it does not take into consideration the users' privacy leakage that such models might produce, since in such approach every OpenID provider knows directly the recommendations given by each user about a particular service provider.

Gal-Oz et al. [66] introduce a number of protocols based on homomorphic encryption, as well as their applicability to trust and reputation management systems. More specifically, it describes a method to aggregate recommendations from end users in distributed systems, in a way that those recommendations remain private. Yet, since none of the entities involved in this process are able to unveil the actual value of each particular recommendation or feedback, but only the aggregation of those, such entities are unable to compare those feedbacks in order to ascertain the similarity between two users.

In turn, authors of [67] apply privacy-preserving collaborative filtering with the aim of predicting the opinion or satisfaction of a given user w.r.t. a particular item in online systems. Such collaborative filtering is sustained on two main pillars, namely: i) previous feedbacks provided by the studied user about other items, and ii) the deviation between the recommendations given by other users about the studied item, and the recommendations given by the studied user about other items. Homomorphic encryption techniques are used in this work as well in order to preserve users' privacy. To this end, authors propose a scenario where several sites have the possibility to share users' recommendations about items, in a privacy-preserving way, in order to obtain predictions based on unknown users. However, this approach, while hiding users' recommendations to external sites, it does not prevent such sites from knowing the actual values of the recommendations from their own users.

In [68], Chow et al. follow on the heel of the previous work and, by using collaborative filtering, aim at developing a customized recommendations system, with the advantage of concealing the feedbacks provided by the end users. Thus, in this approach users submit a recommendation vector to the service providers indicating their feedback or satisfaction for each of the offered services, but introducing some noise consisting of giving fake recommendations about services that they actually did not consume. One of the main motivations driving this solution, from authors' point of view, is to avoid the high costs in terms of performance of a system based on multi-party computation or similar. Nevertheless, despite the noise perturbation techniques in use, this solution would entail a couple of shortcomings for end users: i) service providers would still be able to distinguish and identify those services that users did not consume (those with no feedback at all) and ii) there would be an unavoidable trade-off between privacy and usability when receiving personalized services.

In the same line of supporting context-aware recommendations in order to enable mobile app recommendation and discovery, authors of [69] introduce the Djinn model, a context-aware collaborative filtering algorithm for implicit feedback data based on tensor factorization. While the results shown in this work in terms of customized recommendations for users regarding apps are praiseworthy, the privacy aspects (which, as we have seen, constitute the key for a real deployment and acceptance of these systems by end users) are completely neglected.

Finally, Erkin et al. presented a solution [70] of content-based recommendation algorithms, preserving users' privacy by means of homomorphic encryption schemes. Moreover, the so called "deviations matrix" amongst items is also protected (homomorphically encrypted). Hence, end users can provide their recommendations or feedbacks about certain items to a service provider, by in an encrypted form (so the latter cannot unveil such recommendations). Next, the service provider combines or merges the received encrypted recommendations with the encrypted deviations matrix, without requiring decryption, thanks to the applied homomorphic encryption techniques. Yet, how to initialize or bootstrap such matrices is not clear and might constitute a serious burden nullifying the applicability of the proposed solution in real deployments.

9. Conclusion

Hybrid Broadcast Broadband TV (HbbTV) is a standard aimed to provide features and functionality required to deliver an interactive TV, allowing new entertainment services. One of its main advantages is the possibility to offer vendor applications directly to the users by means of an application store, so the users can browse and install them on their devices.

Due to the large amount of applications which could be deployed, the users need mechanisms to determine which of these applications are worth to install or even if they may include malicious code. In order to help on such decision, reputation management systems are becoming more and more popular. They compute recommendations based on past interactions, in order to predict the behaviour of a given application or service.

Furthermore, users are starting to demand customized reputation values, based on their preferences or on the usage of their services. However, current reputation management systems require the application store to know all the information related to the installed applications and the recommendations provided by each user, hence compromising their privacy.

In this paper we have presented a privacy-preserving reputation framework able to compute customized reputation values, based on the similarity between users. By adding an external trusted entity (identity provider) and using homomorphic encryption techniques, the reputation framework is able to determine the similarity between users from their provided recommendations without revealing such recommendations. In this way, neither the application store nor the identity providers could determine the recommendations provided by a given user or the similarity between two users, yet they can compute personalized reputation values.

In addition, we have performed some tests to analyze the behaviour of the proposed framework in terms of performance. Even though current homomorphic encryption techniques only have been proved to be efficient making operations over bits, we describe some algorithms to extend these techniques so they can compute over natural numbers.

As of future research line directions, we are investigating mechanism and solutions to prevent and avoid a potential collusion between the application store and a malicious or compromised identity provider.

Acknowledgement

This research work has been partially funded by the Next-Generation Hybrid Broadcast Broadband (HBB-NEXT) EU research project, FP7-ICT-2011.1.5, Grant No.: 287848.

References

[1] Hybrid Broadcast Broadband TV (HbbTV), `http://www.hbbtv.org/`, 2013.

[2] S. Jansen, A. Finkelstein, S. Brinkkemper, A sense of community: A research agenda for software ecosystems, in: ICSE Companion, IEEE, 2009, pp. 187–190. doi:10.1109/ICSE-COMPANION.2009.5070978.

[3] J. Slinger, B. Ewoud, Defining app stores: The role of curated marketplaces in software ecosystems, in: ICSOB, 2013, pp. 195–206. doi:10.1007/978-3-642-39336-5_19.

[4] A. Holzer, J. Ondrus, Trends in mobile application development, in: MOBILWARE Workshops, 2009, pp. 55–64. doi:10.1007/978-3-642-03569-2_6.

[5] A. Holzer, J. Ondrus, Mobile application market: A developer's perspective, Telematics and Informatics 28 (2011) 22–31. doi:10.1016/j.tele.2010.05.006.

[6] Y. Zhou, Z. Wang, W. Zhou, X. Jiang, Hey, you, get off of my market: Detecting malicious apps in official and alternative android markets, in: Proceedings of the 19th Annual Network and Distributed System Security Symposium, 2012.

[7] F. Gómez Mármol, G. Martínez Pérez, Providing Trust in Wireless Sensor Networks using a Bio-Inspired Technique, Telecommunication Systems Journal 46 (2011) 163–180. doi:10.1007/s11235-010-9281-7.

[8] F. Gómez Mármol, J. Gómez Marín-Blázquez, G. Martínez Pérez, Linguistic Fuzzy Logic Enhancement of a Trust Mechanism for Distributed Networks, in: Proceedings of the Third IEEE International Symposium on Trust, Security and Privacy for Emerging Applications (TSP-10), Bradford, UK, 2010, pp. 838–845. doi:10.1109/CIT.2010.158.

[9] M. Omar, Y. Challal, A. Bouabdallah, Reliable and fully distributed trust model for mobile ad hoc networks, Computers and Security 28 (2009) 199–214.

[10] Y. Wang, Y. Tao, P. Yu, F. Xu, J. Lu, A Trust Evolution Model for P2P Networks, in: Autonomic and Trusted Computing, number 4610 in LNCS, 4th International Conference, ATC 2007, Springer, Hong Kong, China, 2007, pp. 216–225. doi:10.1007/978-3-540-73547-2_23.

[11] C. Huang, H. Hu, Z. Wang, A dynamic trust model based on feedback control mechanism for P2P applications, in: Autonomic and Trusted Computing, number 4158 in LNCS, Springer, Wuhan, China, 2006, pp. 312–321. doi:10.1007/11839569_30.

[12] Z. Malik, A. Bouguettaya, Reputation bootstrapping for trust establishment among web services, IEEE Internet Computing 13 (2009) 40–47. doi:10.1109/MIC.2009.17.

[13] S. Paradesi, P. Doshi, S. Swaika, Integrating behavioral trust in web service compositions, in: Proceedings of the 2009 IEEE International Conference on Web Services, ICWS '09, 2009, pp. 453–460. doi:10.1109/ICWS.2009.106.

[14] F. Gómez Mármol, G. Martínez Pérez, Security Threats Scenarios in Trust and Reputation Models for Distributed Systems, Elsevier Computers & Security 28 (2009) 545–556. doi:10.1016/j.cose.2009.05.005.

[15] H. K. Kim, J. K. Kim, Y. U. Ryu, Personalized recommendation over a customer network for ubiquitous shopping, Services Computing, IEEE Transactions on 2 (2009) 140–151.

[16] I. Guy, N. Zwerdling, D. Carmel, I. Ronen, E. Uziel, S. Yogev, S. Ofek-Koifman, Personalized recommendation of social software items based on social relations, in: Proceedings of the third ACM conference on Recommender systems, ACM, 2009, pp. 53–60.

[17] P. Resnick, R. Zeckhauser, Trust among strangers in internet transactions: Empirical analysis of ebay's reputation system, The economics of the Internet and E-commerce 11 (2002) 127.

[18] S. S. Shim, G. Bhalla, V. Pendyala, Federated identity management, Computer 38 (2005) 120–122.

[19] T. ElGamal, A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms, in: Proceedings of CRYPTO 84 on Advances in cryptology, 4, 1985, pp. 469–472.

[20] C. Gentry, A fully homomorphic encryption scheme, Ph.D. thesis, Stanford University, 2009.

[21] F. Gómez Mármol, G. Rozinaj, S. Schumann, O. Lábaj, J. Kacur, Smart AppStore: expanding the frontiers of Smartphone ecosystems, IEEE Computer 47 (2014) 42–47. URL: `http://dx.doi.org/10.1109/MC.2014.166`. doi:10.1109/MC.2014.166.

[22] T. Mendyk-Krajewska, Z. Mazur, H. Mazur, Threats to wireless technologies and mobile devices and company network safety, in: Springer (Ed.), Internet - Technical Developments and Applications 2, volume 118 of *Advances in Intelligent and Soft Computing*, 2012, pp. 209–225. doi:10.1007/978-3-642-25355-3_19.

[23] N. Husted, H. Saïdi, A. Gehani, Smartphone security limitations: Conflicting traditions, in: Proceedings of the 2011 Workshop on Governance of Technology, Information, and Policies, GTIP '11, ACM, 2011, pp. 5–12. doi:10.1145/2076496.2076497.

[24] S. Mansfield-Devine, Paranoid android: just how insecure is the most popular mobile platform?, Network Security 2012 (2012) 5–10. doi:10.1016/S1353-4858(12)70081-8.

[25] H. Pieterse, M. S. Olivier, Android botnets on the rise: Trends and characteristics, in: Information Security for South Africa, IEEE, 2012, pp. 1–5. doi:10.1109/ISSA.2012.6320432.

[26] A. P. Felt, M. Finifter, E. Chin, S. Hanna, D. Wagner, A survey of mobile malware in the wild, in: Proceedings of the 1st ACM Workshop on Security and Privacy in Smartphones and Mobile Devices, SPSM '11, ACM, 2011, pp. 3–14. doi:10.1145/2046614.2046618.

[27] Marnix Dekker and Giles Hogben, Appstore Security: 5 lines of defence against malware, 2011.

[28] t. . V. Steve Quirolgico and Jeffrey Voas and Rick Kuhn, IT Professional 13 (2011) 9–11.

[29] Dionysus Blazakis, The Apple Sandbox, 2011.

[30] Sandra Steinbrecher, Enhancing Multilateral Security in and by Reputation Systems, in: proceedings of the IFIP/FIDIS Internet Security and Privacy Summer School, Brno 2008, volume 298, Springer Berlin Heidelberg, 2008, pp. 135–150.

[31] Mina Deng and Kim Wuyts and Riccardo Scandariato and Bart Preneel and Wouter Joosen, A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements, 2010.

[32] R. Kerr, R. Cohen, Smart cheaters do prosper: defeating trust and reputation systems, in: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 2, International Foundation for Autonomous Agents and Multiagent Systems, 2009, pp. 993–1000.

[33] E. Hamilton, M. Kriens, H. Karapandyic, K. Yaici, M. Main, Report on trust and reputation models, 2011.

[34] Randy Farmer and Bryce Glass, Building Web Reputation Systems, O'Reilly Media, 2010.

[35] M. Tavakolifard, K. C. Almeroth, Social computing: an intersection of recommender systems, trust/reputation systems, and social networks, IEEE Network 26 (2012) 53–58. doi:10.1109/MNET.2012.6246753.

[36] L. De Alfaro, A. Kulshreshtha, I. Pye, B. T. Adler, Reputation systems for open collaboration, Commun. ACM 54 (2011) 81–87. doi:10.1145/1978542.1978560.

[37] A. Josang, R. Ismail, C. Boyd, A survey of trust and reputation systems for online service provision, Decision Support Systems 43 (2007) 618–644.

[38] G. Dólera Tormo, F. Gómez Mármol, G. Martínez Pérez, Towards the integration of reputation management in OpenID, Special Issue on Secure Mobility in Future Communication Systems under Standardization, Computer Standards & Interfaces (2013). doi:10.1016/j.csi.2013.08.018.

[39] C. W.-K. Leung, S. C.-F. Chan, F.-L. Chung, G. Ngai, A probabilistic rating inference framework for mining user preferences from reviews, World Wide Web 14 (2011) 187–215.

[40] J. J.-C. Ying, E. H.-C. Lu, W.-C. Lee, T.-C. Weng, V. S. Tseng, Mining user similarity from semantic trajectories, in: Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks, ACM, 2010, pp. 19–26.

[41] G. Dólera Tormo, G. López Millán, G. Martínez Pérez, Definition of an advanced identity management infrastructure, International Journal of Information Security 12 (2013) 173–200. doi:10.1007/s10207-012-0189-y.

[42] A. Jøsang, J. Fabre, B. Hay, J. Dalziel, S. Pope, Trust requirements in identity management, in: Proceedings of the 2005 Australasian workshop on Grid computing and e-research-Volume 44, Australian Computer Society, Inc., 2005, pp. 99–108.

[43] J. Jensen, Benefits of federated identity management: a survey from an integrated operations viewpoint, in: Proceedings of the IFIP WG 8.4/8.9 international cross domain conference on availability, reliability and security for business, enterprise and health information systems, ARES'11, Springer, 2011, pp. 1–12.

[44] R. Canetti, U. Feige, O. Goldreich, M. Naor, Adaptively secure multi-party computation, in: Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing, STOC '96, ACM, New York, NY, USA, 1996, pp. 639–648. URL: http://doi.acm.org/10.1145/237814.238015. doi:10.1145/237814.238015.

[45] W. Du, Z. Zhan, A practical approach to solve secure multi-party computation problems, in: Proceedings of the 2002 Workshop on New Security Paradigms, NSPW '02, ACM, New York, NY, USA, 2002, pp. 127–135. URL: http://doi.acm.org/10.1145/844102.844125. doi:10.1145/844102.844125.

[46] S. Kamara, P. Mohassel, M. Raykova, Outsourcing multi-party computation, Cryptology ePrint Archive, Report 2011/272, 2011. http://eprint.iacr.org/.

[47] M. Naehrig, K. Lauter, V. Vaikuntanathan, Can homomorphic encryption be practical?, in: Proceedings of the 3rd ACM Workshop on Cloud Computing Security Workshop, CCSW '11, ACM, New York, NY, USA, 2011, pp. 113–124. URL: http://doi.acm.org/10.1145/2046660.2046682. doi:10.1145/2046660.2046682.

[48] F. Gómez Mármol, J. Girao, G. Martínez Pérez, TRIMS, a Privacy-aware Trust and Reputation Model for Identity Management Systems, Elsevier Computer Networks Journal 54 (2010) 2899–2912. doi:10.1016/j.comnet.2010.07.020.

[49] A. Mohan, D. M. Blough, AttributeTrust - a framework for evaluating trust in aggregated attributes via a reputation system, in: Proceedings of the 2008 Sixth Annual Conference on Privacy, Security and Trust, 2008, pp. 201–212. doi:10.1109/PST.2008.28.

[50] E. Maler, D. Reed, The venn of identity: Options and issues in federated identity management, Security & Privacy, IEEE (2008).

[51] OASIS Standard, Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) version 2.0, 2005.

[52] D. Recordon, D. Reed, OpenID 2.0: a platform for user-centric identity management, in: Proceedings of the second ACM workshop on Digital identity management, DIM '06, 2006, pp. 11–16. doi:10.1145/1179529.1179532.

[53] Z. Brakerski, C. Gentry, V. Vaikuntanathan, (leveled) fully homomorphic encryption without bootstrapping, in: Proceedings of the 3rd Innovations in Theoretical Computer Science Conference, ACM, 2012, pp. 309–325.

[54] A. López-Alt, E. Tromer, V. Vaikuntanathan, On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption, in: Proceedings of the 44th symposium on Theory of Computing, ACM, 2012, pp. 1219–1234.

[55] I. Damgård, V. Pastro, N. Smart, S. Zakarias, Multiparty computation from somewhat homomorphic encryption, in: Advances in Cryptology–CRYPTO 2012, Springer, 2012, pp. 643–662.

[56] C. Gentry, S. Halevi, Implementing gentry's fully-homomorphic encryption scheme, in: Proceedings of the 30th Annual international conference on Theory and applications of cryptographic techniques: advances in cryptology, EUROCRYPT'11, Springer-Verlag, Berlin, Heidelberg, 2011, pp. 129–148.

[57] J. R. Douceur, J. S. Donath, The sybil attack, in: Proceedings for the 1st International Workshop on Peer-to-Peer Systems (IPTPS '02), 2002, pp. 251–260.

[58] L. Ahn, M. Blum, N. Hopper, J. Langford, Captcha: Using hard ai problems for security, in: E. Biham (Ed.), Advances in Cryptology EUROCRYPT 2003, volume 2656 of *Lecture Notes in Computer Science*, Springer, 2003, pp. 294–311. doi:10.1007/3-540-39200-9_18.

[59] L. von Ahn, M. Blum, J. Langford, Telling humans and computers apart automatically, Commun. ACM 47 (2004) 56–60. URL: http://doi.acm.org/10.1145/966389.966390. doi:10.1145/966389.966390.

[60] G. Dólera Tormo, F. Gómez Mármol, G. Martínez Pérez, ROMEO: ReputatiOn Model Enhancing OpenID Simulator, in: 19th European Symposium on Research in Computer Security (ESORICS), Security & Trust Management Workshop (STM), 2014.

[61] M. Van Dijk, C. Gentry, S. Halevi, V. Vaikuntanathan, Fully homomorphic encryption over the integers, in: Advances in Cryptology–EUROCRYPT 2010, Springer, 2010, pp. 24–43.

[62] J.-S. Coron, D. Naccache, M. Tibouchi, Public key compression and modulus switching for fully homomorphic encryption over the integers, in: Advances in Cryptology–EUROCRYPT 2012, Springer, 2012, pp. 446–464.

[63] Jean-Sebastien Coron and Mehdi Tibouchi, Implementation of the DGHV fully homomorphic encryption scheme, https://github.com/coron/fhe, 2012.

[64] H. Zhao, X. Li, H-trust: A robust and lightweight group reputation system for peer-to-peer desktop grid, in: Distributed Computing Systems Workshops, 2008. ICDCS '08. 28th International Conference on, 2008, pp. 235–240. doi:10.1109/ICDCS.Workshops.2008.96.

[65] G. Dólera Tormo, F. Gómez Mármol, J. Girao, G. Martínez Pérez, Identity Management: In privacy we trust. Bridging the trust gap in e-Health environments, IEEE Security & Privacy (2013). doi:10.1109/MSP.2013.80.

[66] N. Gal-Oz, N. Gilboa, E. Gudes, Schemes for privately computing trust and reputation, in: Trust Management IV - 4th IFIP WG 11.11 International Conference, IFIPTM 2010, 2010, pp. 1–16. doi:10.1007/978-3-642-13446-3_1.

[67] A. Basu, H. Kikuchi, J. Vaidya, Efficient privacy-preserving collaborative filtering based on the weighted slope one predictor, Journal of Internet Services and Information Security 1 (2011) 1–20.

[68] R. Chow, M. A. Pathak, C. Wang, A practical system for privacy-preserving collaborative filtering, in: 12th IEEE International Conference on Data Mining Workshops, ICDM Workshops, IEEE Computer Society, 2012, pp. 547–554. doi:10.1109/ICDMW.2012.84.

[69] A. Karatzoglou, L. Baltrunas, K. Church, M. Böhmer, Climbing the app wall: Enabling mobile app discovery through context-aware recommendations, in: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, CIKM '12, ACM, 2012, pp. 2527–2530. doi:10.1145/2396761.2398683.

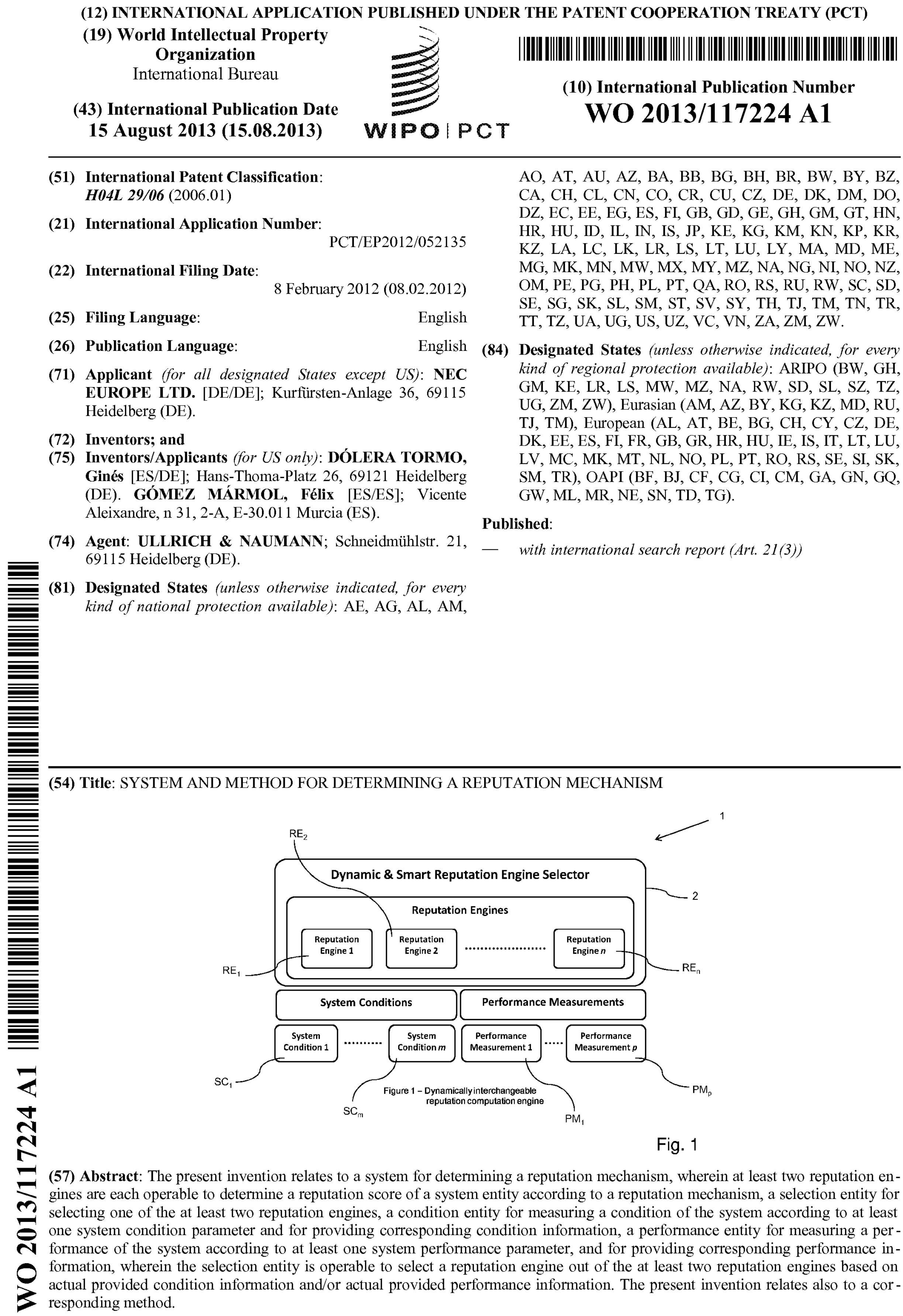
[70] Z. Erkin, M. Beye, T. Veugen, R. L. Lagendijk, Privacy-preserving content-based recommender system, in: Proceedings of the on Multimedia and Security, MM&Sec '12, ACM, 2012, pp. 77–84. doi:10.1145/2361407.2361420.

Appendices

A
Patents

A.I System and method for determining a reputation mechanism

Title:	System and method for determining a reputation mechanism
Authors:	Ginés Dólera Tormo, Félix Gómez Mármol
Type:	International Application Publication under the patent cooperation treaty PCT)
Publication Number:	WO 2013/117224 A1
Year:	2013
Month:	August
Day:	15
State:	Published



(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
15 August 2013 (15.08.2013)

WIPO | PCT

(10) International Publication Number
WO 2013/117224 A1

(51) **International Patent Classification**: *H04L 29/06* (2006.01)

(21) **International Application Number**: PCT/EP2012/052135

(22) **International Filing Date**: 8 February 2012 (08.02.2012)

(25) **Filing Language**: English

(26) **Publication Language**: English

(71) **Applicant** *(for all designated States except US)*: **NEC EUROPE LTD.** [DE/DE]; Kurfürsten-Anlage 36, 69115 Heidelberg (DE).

(72) **Inventors; and**

(75) **Inventors/Applicants** *(for US only)*: **DÓLERA TORMO, Ginés** [ES/DE]; Hans-Thoma-Platz 26, 69121 Heidelberg (DE). **GÓMEZ MÁRMOL, Félix** [ES/ES]; Vicente Aleixandre, n 31, 2-A, E-30.011 Murcia (ES).

(74) **Agent**: **ULLRICH & NAUMANN**; Schneidmühlstr. 21, 69115 Heidelberg (DE).

(81) **Designated States** *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) **Designated States** *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

(54) **Title**: SYSTEM AND METHOD FOR DETERMINING A REPUTATION MECHANISM

Fig. 1

(57) **Abstract**: The present invention relates to a system for determining a reputation mechanism, wherein at least two reputation engines are each operable to determine a reputation score of a system entity according to a reputation mechanism, a selection entity for selecting one of the at least two reputation engines, a condition entity for measuring a condition of the system according to at least one system condition parameter and for providing corresponding condition information, a performance entity for measuring a performance of the system according to at least one system performance parameter, and for providing corresponding performance information, wherein the selection entity is operable to select a reputation engine out of the at least two reputation engines based on actual provided condition information and/or actual provided performance information. The present invention relates also to a corresponding method.

WO 2013/117224 A1

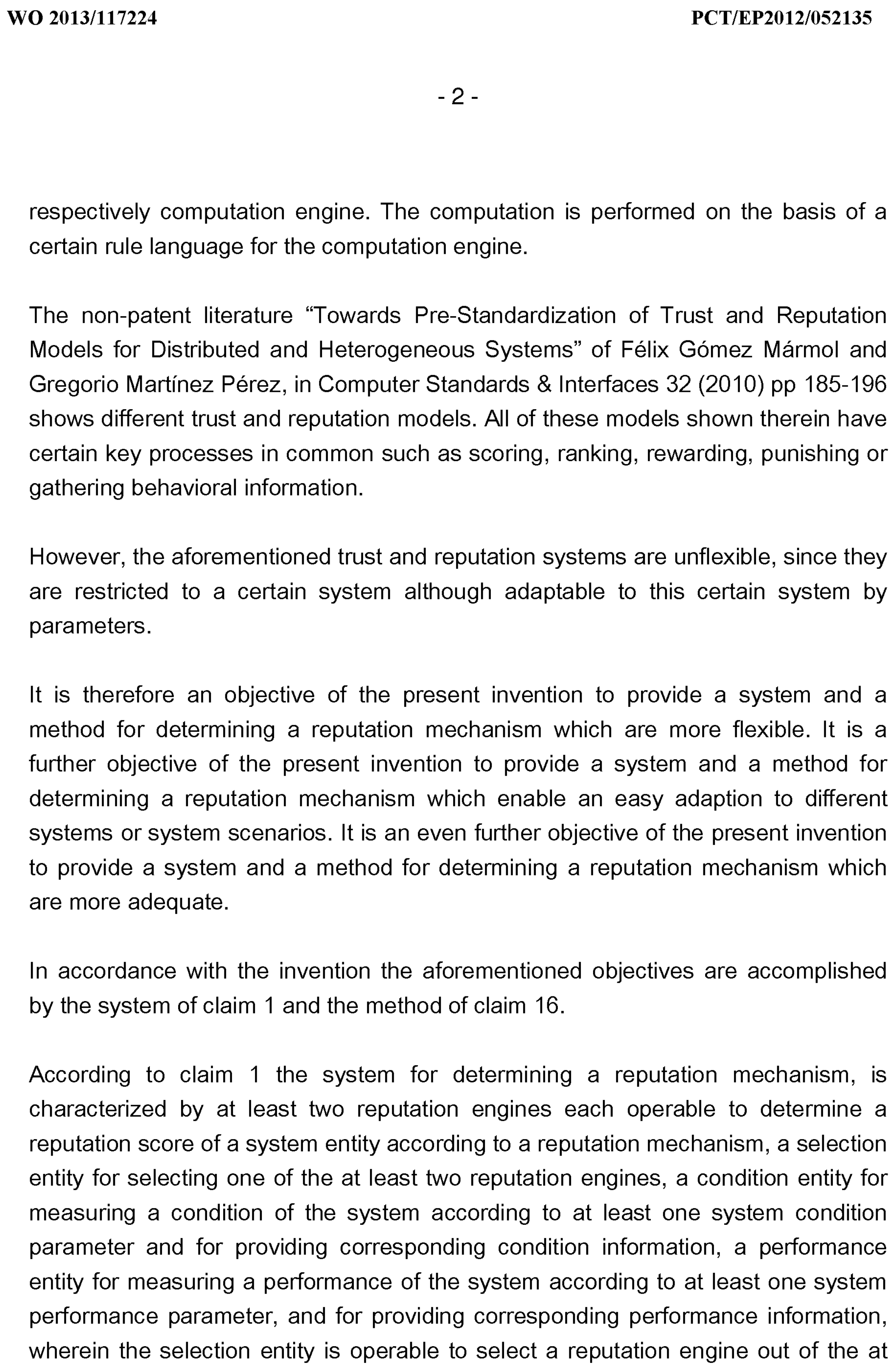
SYSTEM AND METHOD FOR DETERMINING A REPUTATION MECHANISM

The invention relates to a system for determining a reputation mechanism and to a method for determining a reputation mechanism.

Trust and reputation management systems are nowadays widely spread and used in the internet to analyze if certain sources, for example an information provider or the like, are trustworthy, based on experiences of other users, providers, etc.. Such trust or reputation management systems may also be used to determine spam. Other application areas for reputation management systems are ranging from e-commerce ones to blogs, social networks, video streaming services, etc. as well as P2P networks, wireless sensor networks, vehicular ad-hoc networks, cloud computing, identity management systems, collaborative intrusion detection networks, etc..

U.S. 7,937,480 B2 shows a method and a system for operation upon one or more data processors for aggregating reputation data from dispersed reputation engines and for deriving global reputation information. A centralized reputation engine receives feedback from a plurality of local reputation engines, wherein each of the local reputation engines is being operable to determine local reputations based upon of one or more entities and associated with the local reputation engines. An aggregation engine is operable to derive a global reputation for a queried entity based upon an aggregation of the plurality of local reputations. The centralized reputation engine receives reputation queries from one or more of the local reputation engines and applies a local reputation bias to the global reputation based on preferences of the local reputation engine to generate a local reputation for the local reputation engine from the global reputation.

The non-patent literature “Using Reputation to Augment Explicit Authorization” of Philip J. Windley, Devlin Daley, Bryant Cutler and Kevin Tew in the document “proceedings of the 2007 ACN workshop and digital IT management”, ACN, 2007, pp. 271-281, section 4.2.2 shows a model with which a reputation of users may be calculated in a personalized manner using a customizable reputation engine



respectively computation engine. The computation is performed on the basis of a certain rule language for the computation engine.

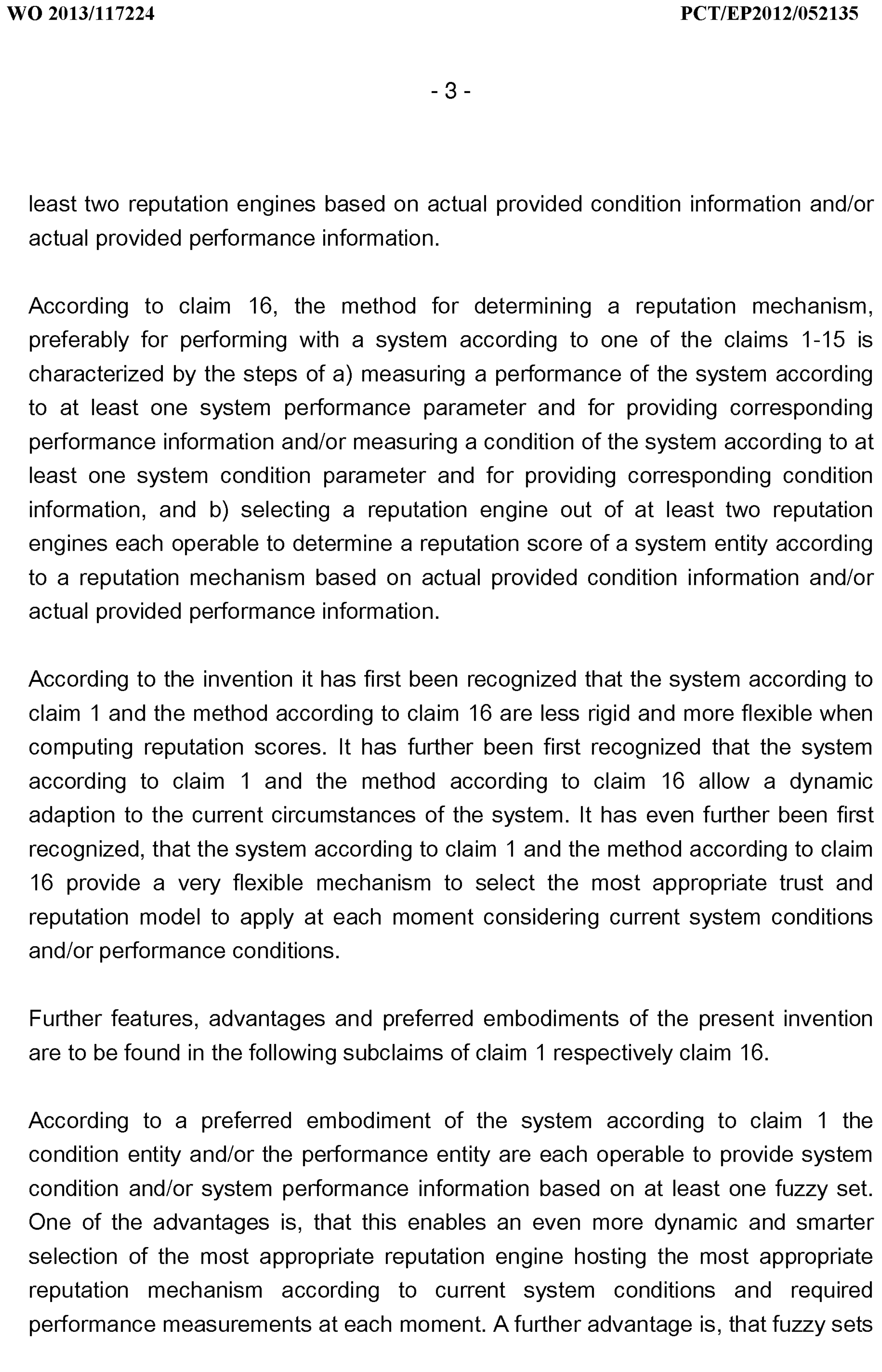
The non-patent literature “Towards Pre-Standardization of Trust and Reputation Models for Distributed and Heterogeneous Systems” of Félix Gómez Mármol and Gregorio Martínez Pérez, in Computer Standards & Interfaces 32 (2010) pp 185-196 shows different trust and reputation models. All of these models shown therein have certain key processes in common such as scoring, ranking, rewarding, punishing or gathering behavioral information.

However, the aforementioned trust and reputation systems are unflexible, since they are restricted to a certain system although adaptable to this certain system by parameters.

It is therefore an objective of the present invention to provide a system and a method for determining a reputation mechanism which are more flexible. It is a further objective of the present invention to provide a system and a method for determining a reputation mechanism which enable an easy adaption to different systems or system scenarios. It is an even further objective of the present invention to provide a system and a method for determining a reputation mechanism which are more adequate.

In accordance with the invention the aforementioned objectives are accomplished by the system of claim 1 and the method of claim 16.

According to claim 1 the system for determining a reputation mechanism, is characterized by at least two reputation engines each operable to determine a reputation score of a system entity according to a reputation mechanism, a selection entity for selecting one of the at least two reputation engines, a condition entity for measuring a condition of the system according to at least one system condition parameter and for providing corresponding condition information, a performance entity for measuring a performance of the system according to at least one system performance parameter, and for providing corresponding performance information, wherein the selection entity is operable to select a reputation engine out of the at



least two reputation engines based on actual provided condition information and/or actual provided performance information.

According to claim 16, the method for determining a reputation mechanism, preferably for performing with a system according to one of the claims 1-15 is characterized by the steps of a) measuring a performance of the system according to at least one system performance parameter and for providing corresponding performance information and/or measuring a condition of the system according to at least one system condition parameter and for providing corresponding condition information, and b) selecting a reputation engine out of at least two reputation engines each operable to determine a reputation score of a system entity according to a reputation mechanism based on actual provided condition information and/or actual provided performance information.

According to the invention it has first been recognized that the system according to claim 1 and the method according to claim 16 are less rigid and more flexible when computing reputation scores. It has further been first recognized that the system according to claim 1 and the method according to claim 16 allow a dynamic adaption to the current circumstances of the system. It has even further been first recognized, that the system according to claim 1 and the method according to claim 16 provide a very flexible mechanism to select the most appropriate trust and reputation model to apply at each moment considering current system conditions and/or performance conditions.

Further features, advantages and preferred embodiments of the present invention are to be found in the following subclaims of claim 1 respectively claim 16.

According to a preferred embodiment of the system according to claim 1 the condition entity and/or the performance entity are each operable to provide system condition and/or system performance information based on at least one fuzzy set. One of the advantages is, that this enables an even more dynamic and smarter selection of the most appropriate reputation engine hosting the most appropriate reputation mechanism according to current system conditions and required performance measurements at each moment. A further advantage is, that fuzzy sets have small memory usage and enable a processing of incomplete and/or approximate data which is suitable for trust and reputation management systems.

According to a further preferred embodiment the condition entity and/or the performance entity are operable to determine at least one condition and/or performance information by using a function mapping a domain with a value for a measured system condition and/or system performance parameter to a range represented by linguistic terms. This enables facilitating an expression of rules and facts. A further advantage is that linguistic terms may be more easily understood by a human user.

According to a further preferred embodiment the selection entity is operable to assign each reputation engine a value representing a suitability of the reputation engine, preferably wherein the range of the value is between 0 and 1 and the sum of the assigned values is 1. One of the advantages of assigning a value representing a suitability of the reputation engine is that the selection of an appropriate reputation engine and further corresponding appropriate reputation mechanism may be easily performed by simply sorting different values and selecting the reputation engine corresponding to the highest value representing the reputation engine with the highest suitability for determining a reputation score for the system. If preferably the range of the value is between 0 and 1 and the sum of the assigned values is 1, this enables an assigning corresponding to a probability of the corresponding reputation engine to be selected by the selection entity.

According to a further preferred embodiment the selection entity is operable to perform a transition from a first reputation engine to a second reputation engine during a transition time. This enables a “smooth” transition avoiding a potential abrupt change in a computed reputation score.

According to a further preferred embodiment the selection entity is operable to use a weighing coefficient for each of the reputation scores determined by the first and second reputation engine when changing from the first reputation engine to the second reputation engine. This enables in an easy and efficient way a “smooth” transition from the first to the second reputation engine by determining a weighted reputation score including the reputation scores of the first and second reputation score.

According to a further preferred embodiment the sum of the weighing coefficients is constant, preferably one. One of the advantages of the sum of weighing coefficients being constant is, that a calculation of a reputation score by both the first reputation engine and the second reputation engine weighted by the weighing coefficients is normalized, therefore providing a fast and easy calculation of a weighted reputation score during the transition time as well as enabling an easy implementation.

According to a further preferred embodiment the selection entity is operable to adapt the weighing coefficients during the transition time. This increases the flexibility during the transition time so that the weighing coefficients and further the calculated reputation score may vary dynamically and be adapted according to for example a predetermined scenario.

According to a further preferred embodiment the selection entity is operable to adapt the weighing coefficients and/or the transition time dependent on actual provided condition information and/or actual provided performance information. This further increases the flexibility of the system when calculating a reputation score by the first and second reputation engine during a transition time when changing from the first reputation engine to the second reputation engine. When adapting the weighing coefficients a very “smooth” transition between the first and second reputation engine when calculating a reputation score is provided. If the transition time is adapted depending on actual provided condition information and/or actual provided performance information, meaning that the transition time may be shortened or extended if necessary to ensure a “smooth” transition.

According to further preferred embodiment the measured system performance and/or the measured system condition is stored in a database. This provides a further increase in flexibility of the system since for example when computing or calculating reputation scores by a reputation engine information, preferably in form recommendations, has to be collected and stored for feeding the reputation engines. Collecting this kind of information may then be done independently of the reputation engine. A further advantage is that different reputation engines may get access to the database to receive required information for computing or calculating reputation scores or reputation values.

According to a further preferred embodiment the database is operable to provide reputation engine information based on different system condition and/or different system performance scenarios. This even further increases the flexibility and reliability since the selection entity may contact the database and receive information about the suitability of different reputation engines for different system condition scenarios and/or different system performance scenarios. The selection entity may also choose a reputation engine which is suited best for current system condition and/or system performance. The time for calculating a reputation score by a selected reputation engine is further reduced.

According to a further preferred embodiment the reputation engine information is based on at least one fuzzy set. One of the advantages is, that this enables an even more dynamic, smarter and easier selection of the most appropriate reputation engine hosting the most appropriate reputation mechanism according to current system conditions and/or system performance measurements at each moment. A further advantage is that the reputation engine information based on at least one fuzzy set requires only a small amount of memory and allows a fast processing of incomplete and/or approximate data which is suitable for trust and reputation management systems.

According to a further preferred embodiment the reputation engine information is determined by using a membership function, mapping a domain with predetermined system condition and/or system performance values to a range represented by linguistic values. This enables facilitating an expression of rules and facts. A further advantage is that linguistic terms by be more easily understood by a human user, for example when implementing the reputation mechanisms in the different reputation engines so that the user may implement a variety of reputation mechanisms in different reputation engines to cover a vast amount of possible system scenarios.

According to a further preferred embodiment the system condition parameter represent a number of entities and/or users of the system, a number of providers, a user and/or entity participation, network resources and/or computer resources. Such a system performance parameter or parameters represent important aspects of a current condition of the system or system scenarios, taking for example into account the number of participants, the number of feedbacks of the participants, feedback storage capabilities, computation capabilities or the like. If the system condition parameter represents a number of users, this defines the amount of end-users, participating in the system. If the system condition parameter represents a number of providers, this defines the amount of providers participating in the system. If the system condition parameter represents user and/or entity participation, this specifies how participative the user entities are within the system, in particular this parameter indicates if the user/entities participating in the system are being active and continually requesting services and providing subsequent feedbacks to make the reputation framework of the system efficiently work. If the system condition parameter represents network resources, this indicates how many network resources, preferably in terms of bandwidth, are present in the system. If the system condition parameter represents computer resources, this indicates how many computer resources, preferably in terms of computational capacity, storage, etc., every provider has on average. These system condition parameters may all be measured regularly.

According to a further preferred embodiment the system performance parameter represents accuracy of a calculated reputation score by a reputation engine, user satisfaction, adaptability, behavior with malicious users and/or entities, and/or behavior with malicious providers. If the system performance parameter represents accuracy, this indicates how similar the computed or calculated reputation score with regard to the actual goodness or behavior of a corresponding relying party is. If the system performance parameter represents user satisfaction, this indicates a similarity between the calculated reputation score provided by the framework or system to a user regarding a concrete relying party and the actual satisfaction of feedback of the user with that specific relying party. If the system performance parameter represents adaptability, this indicates an ability of the reputation engine to quickly and accurately react to sudden changes in the behavior of the relying party, by recalculating an appropriate new reputation score. If the system performance parameter represents behavior with malicious users, this indicates the level of resilience of the analyzed reputation engine with regard to malicious users, providing wrong feedbacks. If the system performance parameter represents behavior with malicious providers, this indicates a level of resilience of an analyzed reputation engine with regard to malicious providers providing wrong recommendations. These system performance parameters may all be measured regularly.

According to a further preferred embodiment of the method according to claim 16 step a) is based on at least one fuzzy set for providing corresponding performance information and/or for providing corresponding condition information. One of the advantages is, that this enables an even more dynamic, smarter and easier selection of the most appropriate reputation engine hosting the most appropriate reputation mechanism according to current system conditions and/or system performance measurements at each moment. A further advantage is that the reputation engine information based on at least one fuzzy set requires only a small amount of memory and enables a fast processing of incomplete and/or approximate data which is suitable for trust and reputation management systems.

According to a further preferred embodiment a measured system performance and/or measured system condition value is mapped via a membership function to a fuzzy set representing a linguistic value. This enables facilitating an expression of rules and facts. A further advantage is that linguistic terms by be more easily understood by a human user, for example when implementing the reputation mechanisms in the different reputation engines so that the user may implement a variety of reputation mechanisms in different reputation engines to cover a vast amount of possible system scenarios.

According to a further preferred embodiment a transition between a first and second reputation engine is performed during a transition time. This enables a “smooth” transition avoiding a potential abrupt change in a computed reputation score.

According to a further preferred embodiment during the transition time a weighing coefficient for each of the reputation scores determined by the first and second reputation engine is used for determining a reputation score. This enables in an easy and efficient way a “smooth” transition from the first to the second reputation engine by determining a weighted reputation score including the reputation scores of the first and second reputation score.

According to a further preferred embodiment the weighing coefficients and/or the transition time are adapted during the transition time according to the transition time, and/or depending on actual provided condition information and/or actual provided performance information. This increases the flexibility during the transition time when changing from the first reputation engine to the second reputation engine so that the weighing coefficients and further the calculated reputation score may vary dynamically and be adapted to for example a predetermined scenario. When adapting the weighing coefficients a very “smooth” transition between the first and second reputation engine when calculating a reputation score is provided. The transition time may be adapted depending on actual provided condition information and/or actual provided performance information, meaning that the transition time may be shortened or extended if necessary to ensure a “smooth” transition.

There are several ways how to design and further developed the teaching of the present invention in an advantageous way. To this end it is to be referred to the patent claim subordinate to patent claim 1 and 16 on the one hand and the following explanation of preferred embodiments of the invention by way of example illustrating by the drawing on the other hand. In connection with the explanation of the preferred example of an embodiment of the invention by the aid of the drawing, generally preferred embodiments and further developments of the teaching will be explained. In the drawings

Fig. 1 shows a schematic view of a system according to a first embodiment;

Fig. 2 shows examples of fuzzy sets to represent system conditions and performance measurements of a system according to a second embodiment;

Fig. 3 shows reputation engine selection probabilities;

Fig. 4a, 4b show a transition from a first reputation engine to a second reputation engine;

Fig. 5 shows an adaption of weighing coefficients during a transition time;

Fig. 6 shows a system according to a second embodiment;

Fig. 7a shows different reputation mechanisms in different reputation engines with system condition analysis;

Fig. 7b shows different reputation mechanisms in different reputation engines with performance measurement analysis.

Fig. 1 shows a schematic view of a system according to a first embodiment.

In Fig. 1 reference number 1 denotes a system for determining a reputation mechanism. The system 1 comprises a selection entity 2 for selecting at least one out of a plurality of reputation engines, denoted with references signs RE_1, RE_2, … RE_n. The system conditions are represented by system condition parameters SC_1, ... SC_m. The system performance is represented by different system performance parameters PM_1, … PM_p. The selection entity 2 selects the most appropriate reputation engine RE_1, RE_2 … RE_n according to the current conditions SC_1, … SC_m of the system, for example in terms of number of users, number of feedbacks, available bandwidth, available storage capacity or the like as well as according to a desired performance by evaluating the measured system performance parameters PM_1, … PM_p, for example accuracy, user’s satisfaction, adaptability, resilience to certain attacks, or the like.

Fig. 2 shows examples of fuzzy sets to represent system conditions and performance measurements of a system according to a second embodiment.

In Fig. 2 are shown examples of fuzzy sets to represent system conditions and performance measurements. The selection entity 2 uses several separate membership functions and a fuzzy set for each system condition SC_1, … SC_m and for each performance measurement PM_1, … PM_p. The fuzzy sets $FSSC_1$, … $FSSC_m$ for the system condition and the fuzzy sets $FSPM_1$, … $FSPM_p$ for the performance measurement comprise each linguistic terms LT_1, LT_2, LT_3, in particular in Fig. 2 the linguistic terms “low”, “medium” and “high”, and these are defining particular ranges. The corresponding membership functions map the values for the specific system condition parameters SC_1, … SC_m and the measured specific system performance parameters PM_1, … PM_p to the corresponding fuzzy sets $FSSC_1$, … $FSSC_m$, $FSPM_1$, … $FSPM_p$ and further to the linguistic terms LT_1, LT_2, LT_3 forming the fuzzy sets $FSSC_1$, … $FSSC_m$, $FSPM_1$, … $FSPM_p$.

The selection entity 2 now gets a set of labels, representing the current condition and the current performance of the system, for instance such a system condition may indicates the number of users being “low” and the user participation is classified as “medium”, etc.. The selection entity 2 then determines according to this provided information the suitability of each of the reputation engines RE_1, RE_2, …, RE_p as a value between 0 and 1 in such a way that the sum of all these values is equal to 1. These values now represent the probability of a use of each reputation engine RE_1, RE_2, …, RE_p for calculating a reputation score:

$$f(SC_1, SC_2, \ldots, SC_m, PM_1, PM_2, \ldots, PM_p) = (p_{RE_1}, p_{RE_2}, \ldots, p_{RE_n})$$

where SC_i represents the i-th system condition, PM_j represents the j-th performance measurement, RE_k represents the k-th reputation engine and p_{RE_k} the probability of choosing this reputation engine as the current applied one. Moreover, as mentioned before:

$$\sum_{k=1}^{n} p_{RE_k} = 1$$

is satisfied.

Fig. 3 shows reputation engine selection probabilities.

Fig. 3 shows an example of how the probabilities of the reputation engines RE_1, RE_2, … RE_n may look like: the probability for reputation engine p_{RE_1} to be chosen by the selection entity 2 is equal to the probability of reputation engine p_{RE_2} and the probability for selecting reputation engine p_{RE_4}. The probability for reputation engine p_{RE_3} is smaller than those for reputation engines RE_1, RE_2 and RE_4.

Fig. 4a, 4b show a transition from a first reputation engine to a second reputation engine.

In Fig. 4a an abrupt change in a computed reputation score R is shown over time t when a new reputation engine, here RE_2, is selected and the reputation score R was previously calculated by reputation engine RE_1. Such a change or transition between two different reputation engines RE_1, RE_2 may be required, if a probability of a current reputation engine decreases below a pre-given threshold level, or when certain system condition and/or certain performance parameters change too fast and/or too much.

Fig. 4b shows a “smooth” transition of the reputation score R when changing the reputation engine from reputation engine RE_1 to reputation engine RE_2. In order to achieve this smooth transition, once the reputation engine exchange has been triggered by the selection entity 2 the complete transition will be prolonged for a certain transition time t_T. If the reputation engine RE_1 is the current reputation engine and reputation engine RE_2 is the selected reputation engine to replace the current reputation engine RE_1, during the transition time t_T the reputation score R will be computed, taking into account the outputs respecitively the reputation scores R_{RE_1}, R_{RE_2} from both reputation engines RE_1, RE_2 by weighing them: the calculated reputation R is then be calculated in the following way:

$$R_{transition\_time} = C_1 \cdot R_{RE_1} + C_2 \cdot R_{RE_2}$$

These weighing coefficients C_1 and C_2 which fulfill $C_1 + C_2 = 1$ will be dynamically adapted in a manner that at the beginning of the transition time t_T the first weighing coefficient C_1 is equal to 1 and the second weighing coefficient C_2 is equal to 0 and at the end of the transition time the first weighing coefficient C_1 is 0 and the second weighing coefficient is 1.

The transition time t_T may be also adapted meaning shortened or extended according to system conditions and/or performance measurements.

Fig. 5 shows an adaption of weighing coefficients during a transition time.

In Fig. 5 is shown a steady non - linear decrease of weighing coefficient C_1 from the beginning of the transition time t_T from 1 to 0 at the end of the transition time t_T and the steady increase of the weighing coefficient C_2 starting at the value of 0 at the beginning of the transition time t_T to a value of 1 at the end of the transition time t_T. The weighing coefficient C_1 is decreased in the same portion as the second weighing coefficient C_2 is increased so that $C_1 + C_2 = 1$ is fullfilled. Both weighing coefficients C_1, C_2 and the duration of the transition time t_T may also be adapted in particular according to system conditions and/or system performance.

Fig. 6 shows a system according to a second embodiment.

In Fig. 6 a system 1 is shown with a relying party RP, different open ID providers P_1, P_2, P_3, P_4, and a user UE.

If the user UE wants to get access to a certain service, the user UE may contact a relying party RP. However, the user UE would like to clarify if the relying party RP is trustworthy. The user UE therefore authenticates with a first Open ID provider P_1 to request a reputation for the relying party RP from its open ID provider P_1. The Open ID provider P_1 collects recommendations for the relying party RP from for example other Open ID providers P_2, P_3, P_4 each identifying the relying party RP by authentication. The other Open ID providers P_2, P_3, P_4 may further collect recommendations for the identified relying party RP by other users connected to the other open ID providers P_2, P_3, P_4. To calculate a reputation score of the relying party RP the Open ID provider P_1 – as already mentioned – collects the recommendations for the relying party of the other Open ID providers P_2, P_3, P_4 and may store the collected recommendations in a database D connected to the Open ID provider P_1. The Open ID provider P_1 may then calculate a reputation score of the target relying party RP by one of the following four models: The first model calculates or computes an

average for the reputation score R

$$R \quad (RP) = \alpha\left(\frac{1}{n}\sum_{i=1}^{n} Rec_{u_i}\right) + \beta\left(\frac{1}{m}\sum_{i=1}^{m} Rec_{op_i}\right)$$

where R(RP) is reputation score of the target relying party RP,
α is weight of the user's recommendations
β is weight of the OpenID Provider recommendation
n is number of user recommendations
m is number of OpenID Provider recommendations
Rec_{u_i} is aggregation of recommendations given by the *i*-th user
Rec_{op_i} is last recommendation given by the *i*-th OpenID Provider

The second model calculates a weighted average for the reputation score

$$R \quad (RP)_t = \alpha\left(\frac{\sum_{i=1}^{n} w_{u_{i,t}} \cdot Rec_{u_i}}{\sum_{i=1}^{n} w_{u_{i,t}}}\right) + \beta\left(\frac{\sum_{i=1}^{n} w_{op_{i,t}} \cdot Rec_{op_i}}{\sum_{i=1}^{n} w_{op_{i,t}}}\right)$$

where

R(RP)$_t$ is reputation score of the target relying party RP, at time *t*,

$w_{u_{i,t}} = f\left(w_{u_{i,t-1}}, \left|Rep - Rec_{u_i}\right|\right)$ is a weight given to $\mathrm{Re}c_{u_i}$ at time *t*,

$w_{op_{i,t}} = f\left(w_{op_{i,t-1}}, \left|Rep - Rec_{op_i}\right|\right)$ is a weight given to $\mathrm{Re}c_{op_i}$ at time *t*,

The third model calculates a preferences weighted average

$$R \quad (RP)_{j,t} = \alpha\left(\frac{\sum_{i=1}^{n} w_{u_{i,t}} \cdot sim_{i,j} \cdot Rec_{u_i}}{\sum_{i=1}^{n} w_{u_{i,t}} \cdot sim_{i,j}}\right) + \beta\left(\frac{\sum_{i=1}^{n} w_{op_{i,t}} \cdot Rec_{op_i}}{\sum_{i=1}^{n} w_{op_{i,t}}}\right)$$

where
R(RP)$_{j,\,t}$ is reputation score of the target relying party RP, at time t, customized for user j,

$sim_{i,j} = \sigma\left(pref_i, pref_j\right)$ is representing a similarity between users *i* and *j*, based on the similarity between their service provision preferences (delivery time, quality, price, etc).

The forth model calculates a users weighted average

$$R \quad (RP)_{j,t} = \alpha\left(\frac{\sum_{i=1}^{n} w_{u_{i,j,t}} \cdot Rec_{u_i}}{\sum_{i=1}^{n} w_{u_{i,j,t}}}\right) + \beta\left(\frac{\sum_{i=1}^{n} w_{op_{i,j,t}} \cdot Rec_{op_i}}{\sum_{i=1}^{n} w_{op_{i,j,t}}}\right)$$

where

$w_{u_{i,j,t}} = f\left(w_{u_{i,j,t-1}}, \left|Rec_{u_i} - Rec_{u_j}\right|\right)$ is a weight given by user *j* to Rec_{u_i} at time *t*, and

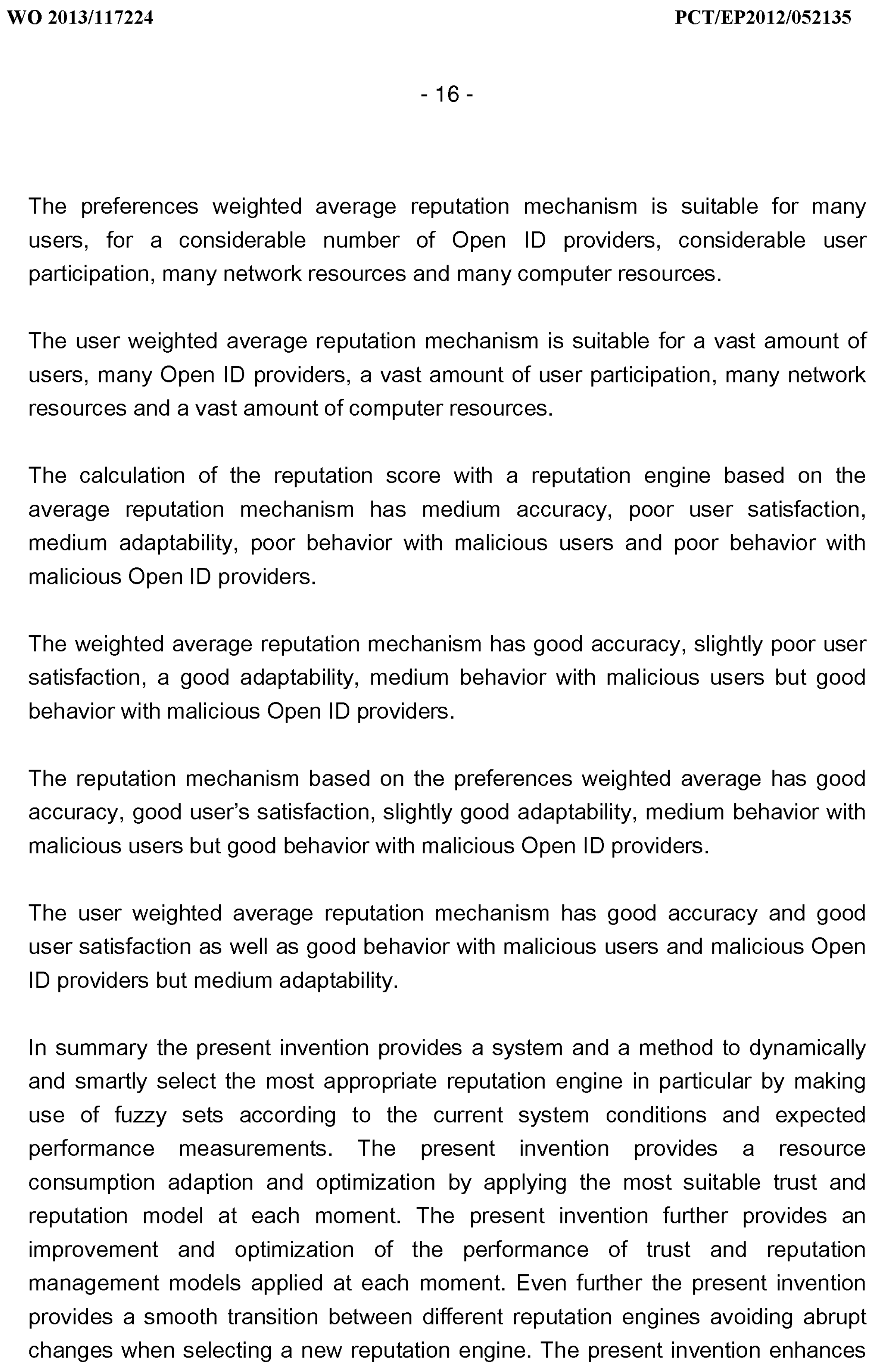
$w_{op_{i,j,t}} = f\left(w_{op_{i,j,t-1}}, \left|Rec_{op_i} - Rec_{u_j}\right|\right)$ is a weight given by user *j* to Rec_{op_i} at time *t*

After calculation of the reputation score R(RP) of the target relying party RP according to one of the afore mentioned models, the user UE gets the reputation score R(RP) for the relying party RP from the Open ID provider P_1 and may decide, if the relying party RP is trustworthy to get or receive the requested service.

Fig. 7a shows different reputation mechanisms in corresponding reputation engines with system condition analysis. Fig. 7b shows different reputation mechanisms in different reputation engines with performance measurement analysis.

Beforehand the aforementioned four models for a reputation mechanism and calculated on a reputation engine have been analyzed to certain system conditions and performance measurements which are laid down in Fig. 7a for system conditions and in Fig. 7b for performance measurements. For example if the reputation score is calculated by the reputation engine with the reputation mechanism based on the average model, this model is suitable for a few number of users, a few number of Open ID providers, few user participation, few network resources and few computer resources.

The weighted average reputation mechanism is suitable for a considerable number of users, for a few number of Open ID providers, for considerable user participation, for few network resources and considerable computer resources.



The preferences weighted average reputation mechanism is suitable for many users, for a considerable number of Open ID providers, considerable user participation, many network resources and many computer resources.

The user weighted average reputation mechanism is suitable for a vast amount of users, many Open ID providers, a vast amount of user participation, many network resources and a vast amount of computer resources.

The calculation of the reputation score with a reputation engine based on the average reputation mechanism has medium accuracy, poor user satisfaction, medium adaptability, poor behavior with malicious users and poor behavior with malicious Open ID providers.

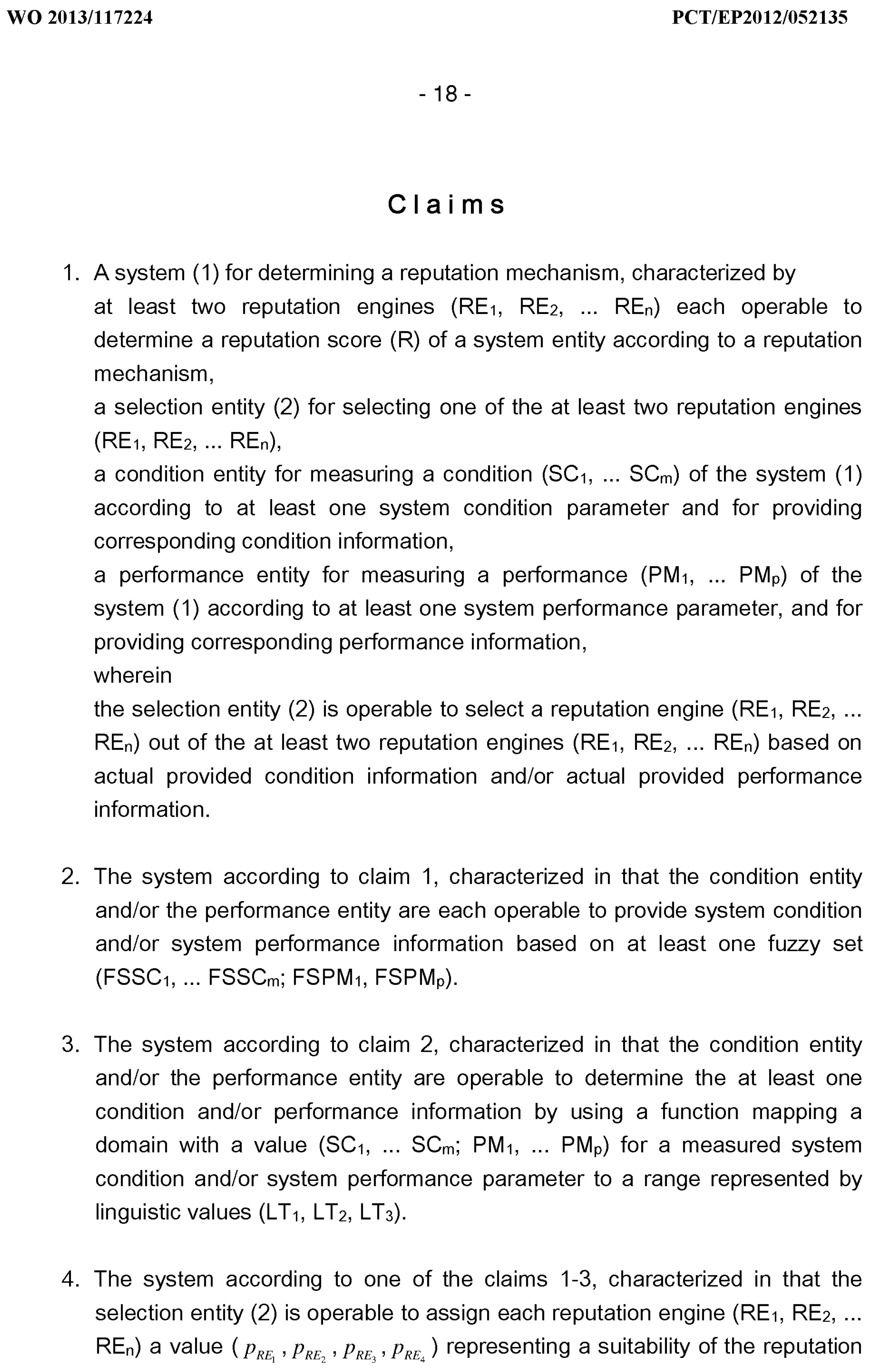
The weighted average reputation mechanism has good accuracy, slightly poor user satisfaction, a good adaptability, medium behavior with malicious users but good behavior with malicious Open ID providers.

The reputation mechanism based on the preferences weighted average has good accuracy, good user's satisfaction, slightly good adaptability, medium behavior with malicious users but good behavior with malicious Open ID providers.

The user weighted average reputation mechanism has good accuracy and good user satisfaction as well as good behavior with malicious users and malicious Open ID providers but medium adaptability.

In summary the present invention provides a system and a method to dynamically and smartly select the most appropriate reputation engine in particular by making use of fuzzy sets according to the current system conditions and expected performance measurements. The present invention provides a resource consumption adaption and optimization by applying the most suitable trust and reputation model at each moment. The present invention further provides an improvement and optimization of the performance of trust and reputation management models applied at each moment. Even further the present invention provides a smooth transition between different reputation engines avoiding abrupt changes when selecting a new reputation engine. The present invention enhances trust and reputation systems in such a way that they can be dynamically adapted to the system conditions depending also on the targeted performance metrics. The present invention avoids designing and developing tunable trust and reputation models for each specific scenario but provides rather models having a high performance and a certain configuration of system conditions and/or performance measurements. The present invention further enhances end user experience in the trust and reputation management systems by means of applying the best performing reputation model at each moment.

Many modifications and other embodiments of the invention set forth herein will come to mind the one skilled in the art to which the invention pertains having the benefit of the teachings presented in the foregoing description and the associated drawings. Therefore it is to be understood that the invention is not to be limited to the specific embodiments disclosed and that modifications and other embodiments are intended to be included within the scope of the appended claims. Although specific claims are employed herein, there are used in generic and descriptive sense only and not for purposes of limitation.



WO 2013/117224 PCT/EP2012/052135

- 18 -

Claims

1. A system (1) for determining a reputation mechanism, characterized by at least two reputation engines (RE_1, RE_2, ... RE_n) each operable to determine a reputation score (R) of a system entity according to a reputation mechanism,
   a selection entity (2) for selecting one of the at least two reputation engines (RE_1, RE_2, ... RE_n),
   a condition entity for measuring a condition (SC_1, ... SC_m) of the system (1) according to at least one system condition parameter and for providing corresponding condition information,
   a performance entity for measuring a performance (PM_1, ... PM_p) of the system (1) according to at least one system performance parameter, and for providing corresponding performance information,
   wherein
   the selection entity (2) is operable to select a reputation engine (RE_1, RE_2, ... RE_n) out of the at least two reputation engines (RE_1, RE_2, ... RE_n) based on actual provided condition information and/or actual provided performance information.

2. The system according to claim 1, characterized in that the condition entity and/or the performance entity are each operable to provide system condition and/or system performance information based on at least one fuzzy set ($FSSC_1$, ... $FSSC_m$; $FSPM_1$, $FSPM_p$).

3. The system according to claim 2, characterized in that the condition entity and/or the performance entity are operable to determine the at least one condition and/or performance information by using a function mapping a domain with a value (SC_1, ... SC_m; PM_1, ... PM_p) for a measured system condition and/or system performance parameter to a range represented by linguistic values (LT_1, LT_2, LT_3).

4. The system according to one of the claims 1-3, characterized in that the selection entity (2) is operable to assign each reputation engine (RE_1, RE_2, ... RE_n) a value (p_{RE_1} , p_{RE_2} , p_{RE_3} , p_{RE_4}) representing a suitability of the reputation

engine (RE_1, RE_2, ... RE_n), preferably wherein the range of the value is between 0 and 1 and the sum of the assigned value ($p_{RE_1}, p_{RE_2}, p_{RE_3}, p_{RE_4}$) is 1.

5. The system according to one of the claims 1-4, characterized in that the selection entity (2) is operable to perform a transition from a first reputation engine (RE_1) to a second reputation engine (RE_2) during a transition time (t_T).

6. The system according to one of the claims 1-5, characterized in that the selection entity (2) is operable to use a weighing coefficient (C_1, C_2) for each of the reputation scores (R) determined by the first and second reputation engine (RE_1, RE_2) when changing from the first reputation engine (RE_1) to the second reputation engine (RE_2).

7. The system according to claim 6, characterized in that the sum of the weighing coefficients (C_1, C_2) is constant, preferably one.

8. The system according to one of the claims 5-7, characterized in that the selection entity (2) is operable to adapt the weighing coefficients (C_1, C_2) during the transition time (t_T).

9. The system according to one of the claims 5-8, characterized in that the selection entity (2) is operable to adapt the weighing coefficients (C_1, C_2) and/or the transition time (t_T) dependant on actual provided condition information and/or actual provided performance information.

10. The system according to one of the claims 1-9, characterized in that the measured system performance and/or the measured system condition is stored in a database (D).

11. The system according to one of the claims 1-10, characterized in that the database (D) is operable to provide reputation engine information based on different system condition and/or different system performance scenarios.

12. The system according to claim 11, characterized in that the reputation engine information is based on at least one fuzzy set ($FSSC_1$, $FSSC_m$, $FSPM_1$, ... , $FSPM_p$).

13. The system according to claim 12, characterized in that the reputation engine information is determined by using a membership function mapping a domain with predefined system condition and/or system performance values (SC_1, ... SC_m; PM_1, ... PM_p) to a range represented by linguistic values (LT_1, LT_2, LT_3).

14. The system according to one of the claims 1-13, characterized in that the system condition parameter represent a number of entities (P_1, P_2, P_3, P_4, UE, RP) and/or users of the system, a number of providers (P_1, P_2, P_3, P_4,) a user and/or entity participation, network resources and/or computer resources.

15. The system according to one of the claims 1-14, characterized in that the system performance parameter represents accuracy of a calculated reputation score by a reputation engine (RE_1, RE_2, ... RE_n), user satisfaction, adaptability, behaviour with malicious users and/or entities, and/or behaviour with malicious providers.

16. A method for determining a reputation mechanism, preferably for performing with a system according to one of the claims 1-15, characterized by the steps of
    a) Measuring a performance of the system (1) according to at least one system performance parameter and for providing corresponding performance information and/or measuring a condition of the system (1) according to at least one system condition parameter and for providing corresponding condition information, and
    b) Selecting a reputation engine (RE_1, RE_2, ... RE_n) out of at least two reputation engines (RE_1, RE_2, ... RE_n) each operable to determine a reputation of a system entity (RP) according to a reputation mechanism based on actual provided condition information and/or actual provided performance information.

17. The method according to claim 16, characterized in that step a) is based on at least one fuzzy set ($FSSC_1$, ... $FSSC_m$; $FSPM_1$, ... $PSPM_p$) for providing corresponding performance information and/or for providing corresponding condition information.

18. The method according to one of the claims 16-17, characterized in that a measured system performance and/or measured system condition value (SC_1, ... SC_m; PM_1, ... PM_p) is mapped via a membership function to a fuzzy set ($FSSC_1$, ... $FSSC_m$; $FSPM_1$, ... $PSPM_p$) representing a linguistic value (LT_1, LT_2, LT_3).

19. The method according to one of the claims 16-18, characterized in that a transition between a first and second reputation engine (RE_1, RE_2) is performed during a transition time (t_T).

20. The method according to one of the claims 16-19, characterized in that during the transition time (t_T) a weighing coefficient (C_1, C_2) for each of the reputation scores determined by the first and second reputation engine (RE_1, RE_2) is used for determining a reputation score.

21. The method according to claim 20, characterized in that the weighing coefficients (C_1, C_2) and/or the transition time (t_T) are adapted during the transition time (t_T), according to the transition time (t_T) and/or dependant on actual provided condition information and/or actual provided performance information.

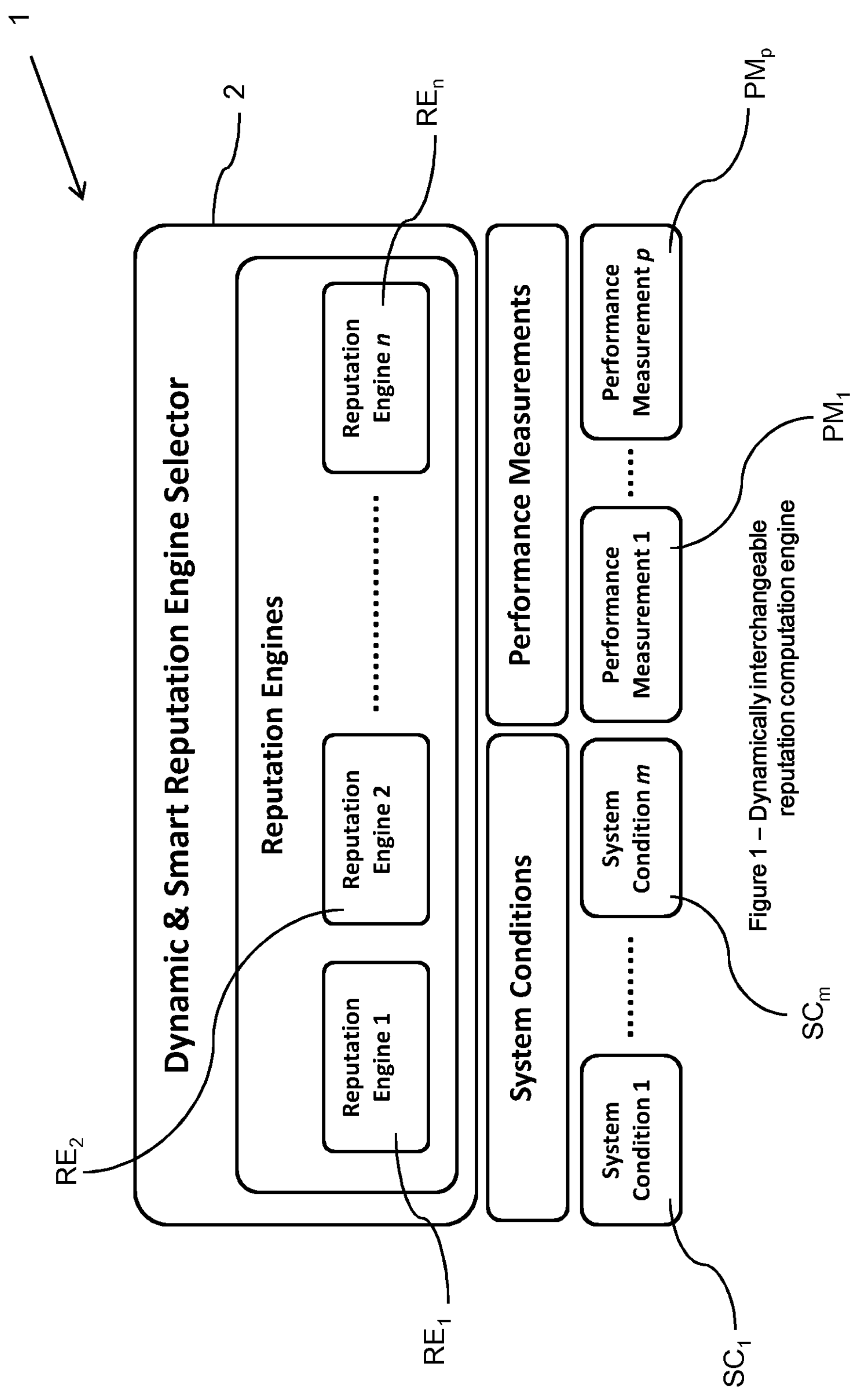


Figure 1 – Dynamically interchangeable reputation computation engine

Fig. 1

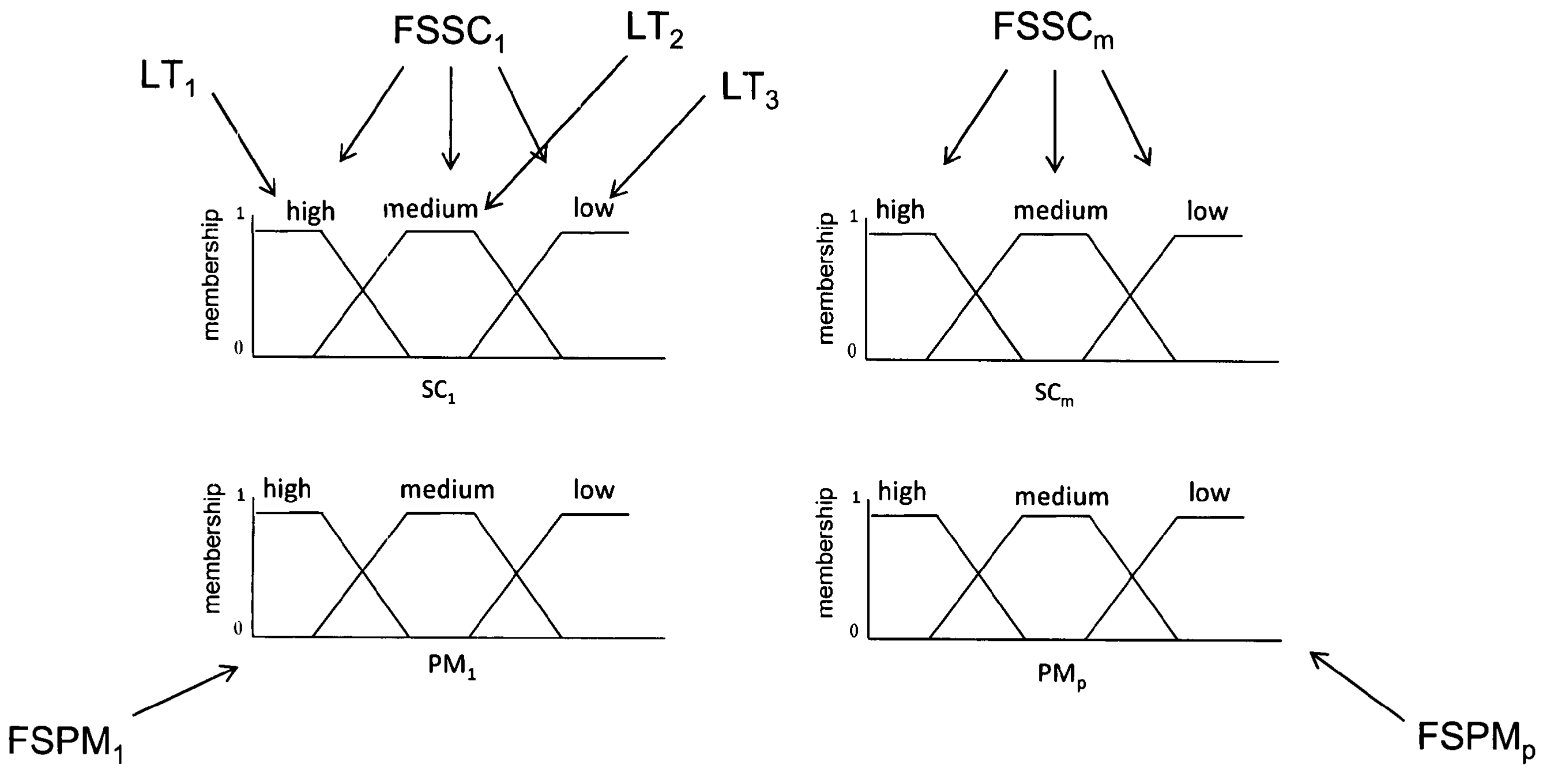


Figure 2 – Example of fuzzy sets to represent system conditions and performance measurements

Fig. 2

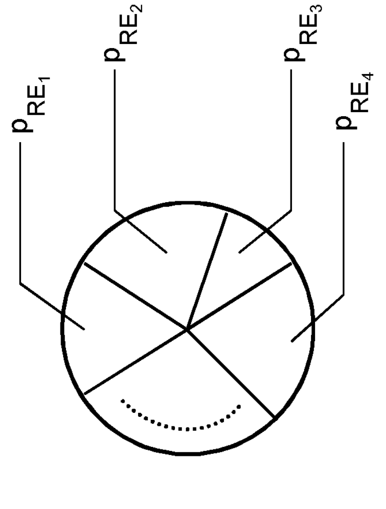


Figure 3 – Reputation computation engines selection probabilities

Fig. 3

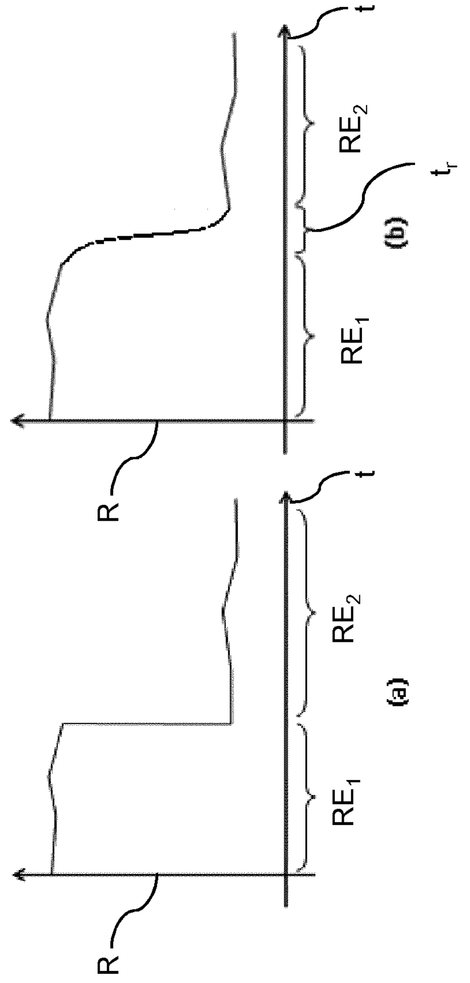


Fig. 4

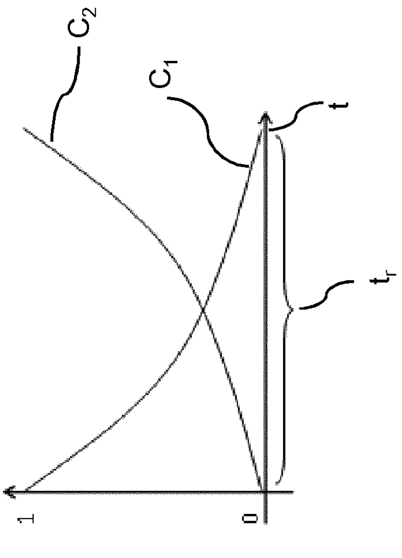


Figure 5 – C$_1$ and C$_2$ adaptation during transition time

Fig. 5

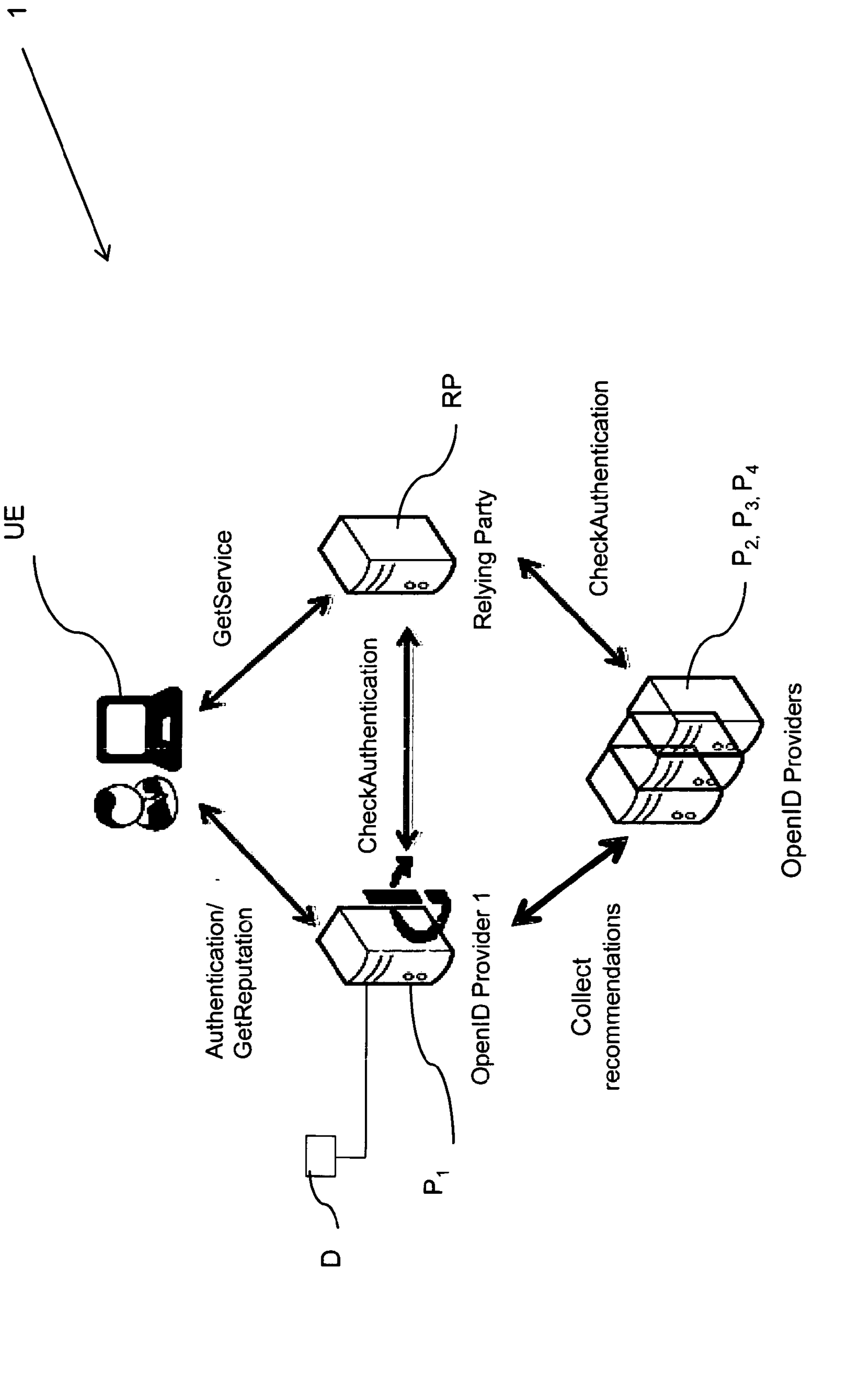
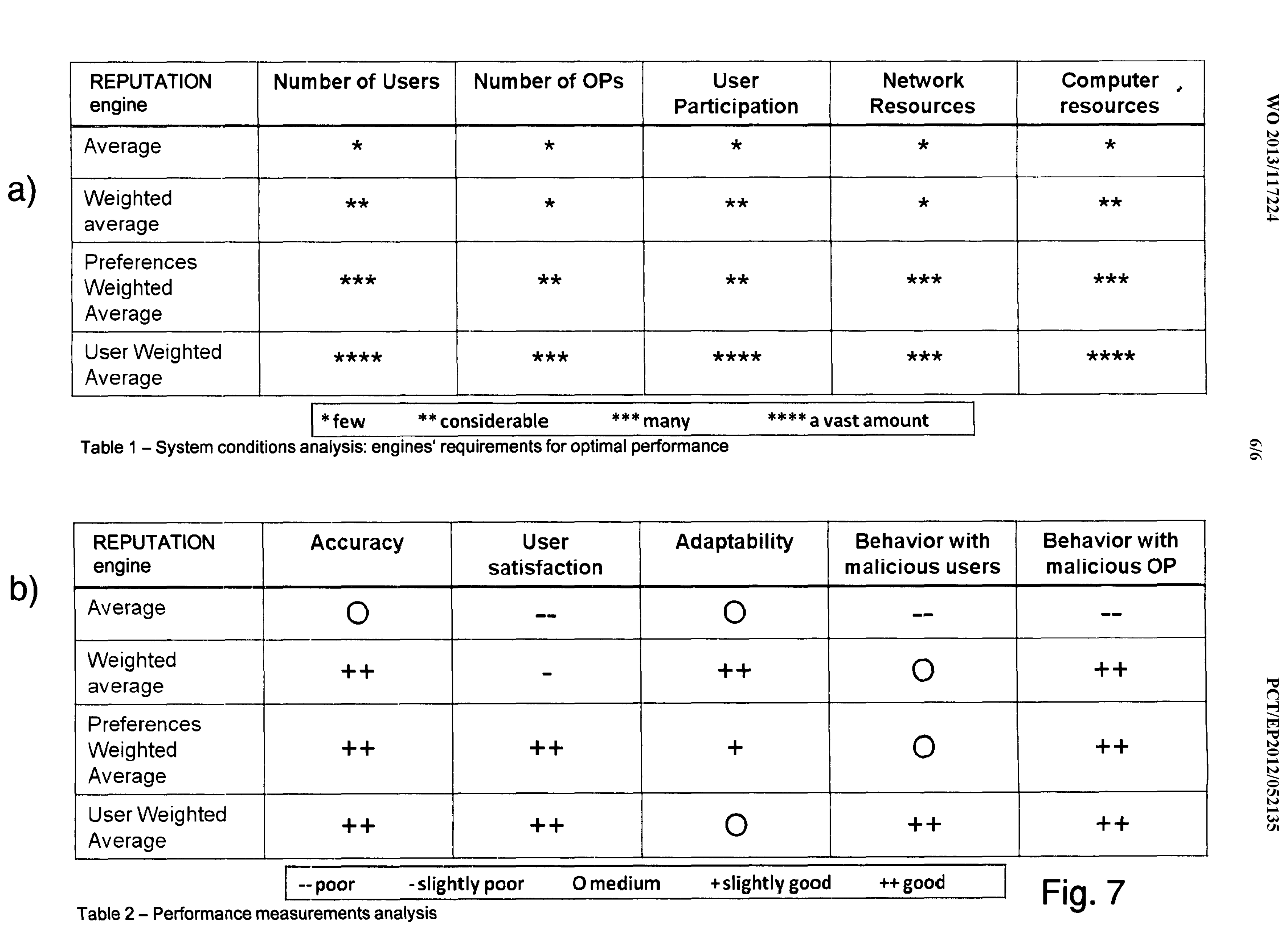


Fig. 6 – Reputation management integration into OpenID

Fig. 6



WO 2013/117224 6/6 PCT/EP2012/052135

a)

REPUTATION engine	Number of Users	Number of OPs	User Participation	Network Resources	Computer resources
Average	*	*	*	*	*
Weighted average	**	*	**	*	**
Preferences Weighted Average	***	**	**	***	***
User Weighted Average	****	***	****	***	****

* few	** considerable	*** many	**** a vast amount

Table 1 – System conditions analysis: engines' requirements for optimal performance

b)

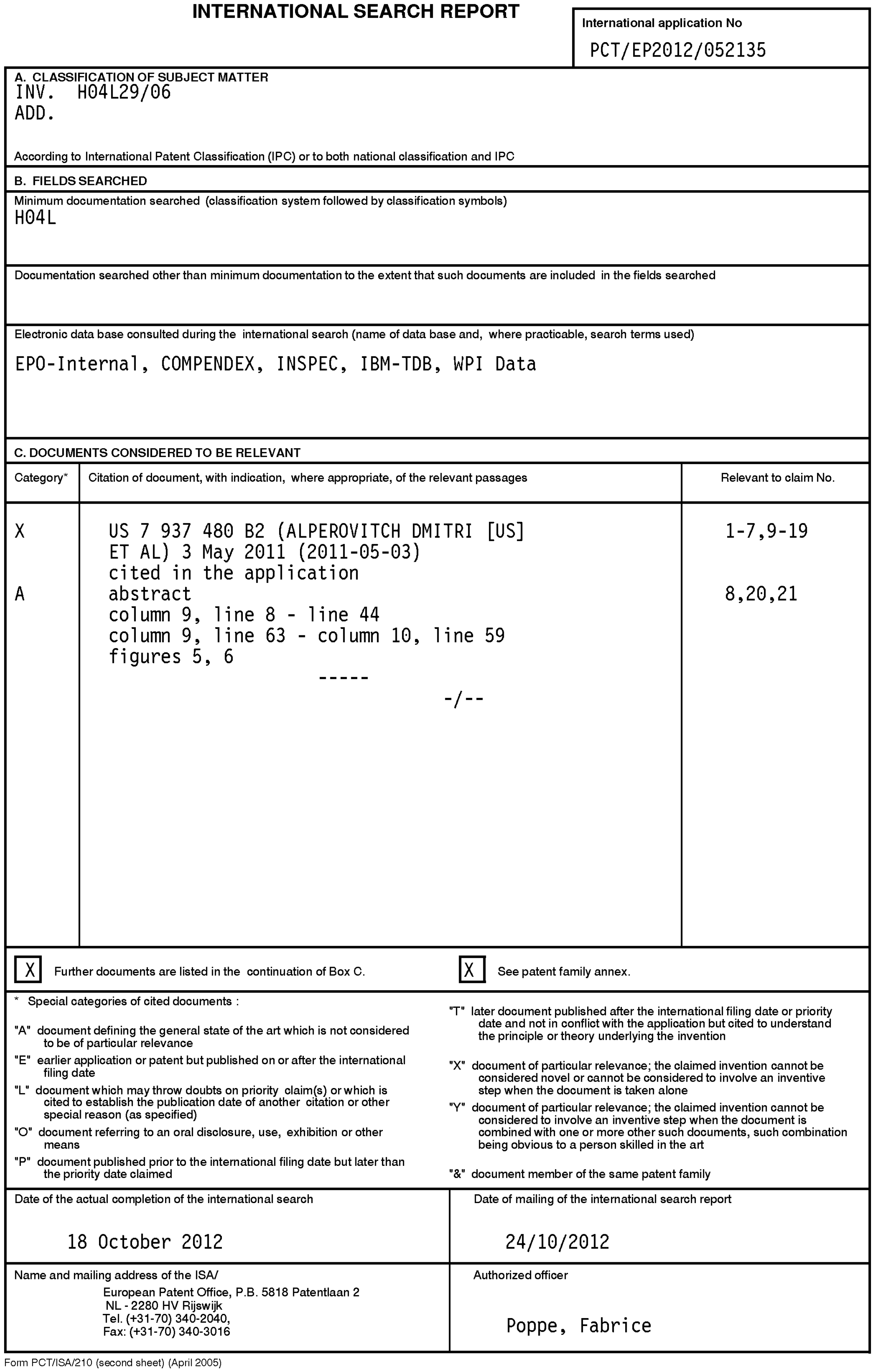
REPUTATION engine	Accuracy	User satisfaction	Adaptability	Behavior with malicious users	Behavior with malicious OP
Average	O	--	O	--	--
Weighted average	++	-	++	O	++
Preferences Weighted Average	++	++	+	O	++
User Weighted Average	++	++	O	++	++

-- poor	- slightly poor	O medium	+ slightly good	++ good

Table 2 – Performance measurements analysis

Fig. 7

SUBSTITUTE SHEET (RULE 26)



INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2012/052135

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04L29/06
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal, COMPENDEX, INSPEC, IBM-TDB, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 7 937 480 B2 (ALPEROVITCH DMITRI [US] ET AL) 3 May 2011 (2011-05-03) cited in the application	1-7,9-19
A	abstract column 9, line 8 - line 44 column 9, line 63 - column 10, line 59 figures 5, 6 ----- -/--	8,20,21

☒ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

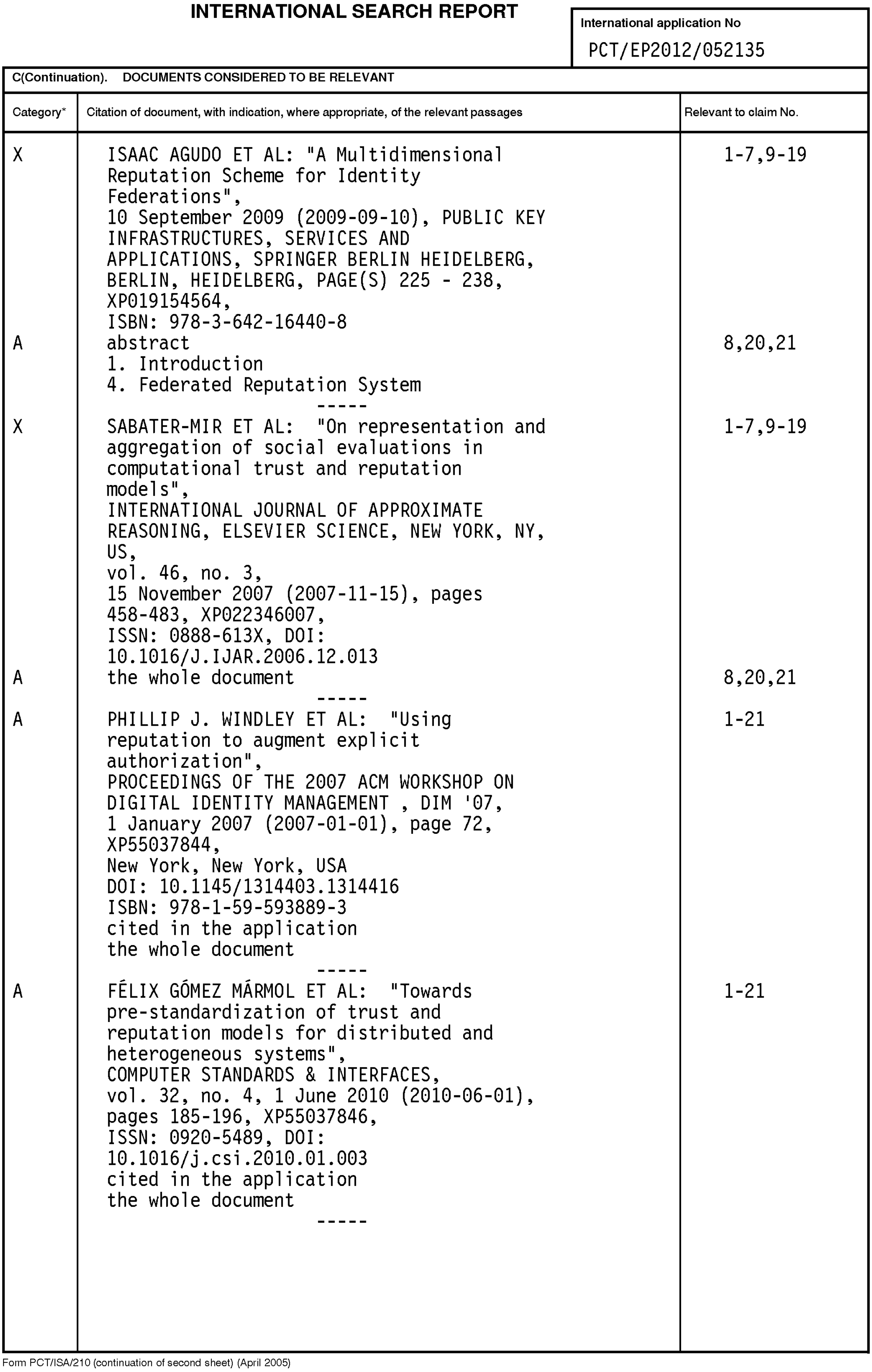
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search 18 October 2012	Date of mailing of the international search report 24/10/2012
Name and mailing address of the ISA/ European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Authorized officer Poppe, Fabrice

Form PCT/ISA/210 (second sheet) (April 2005)



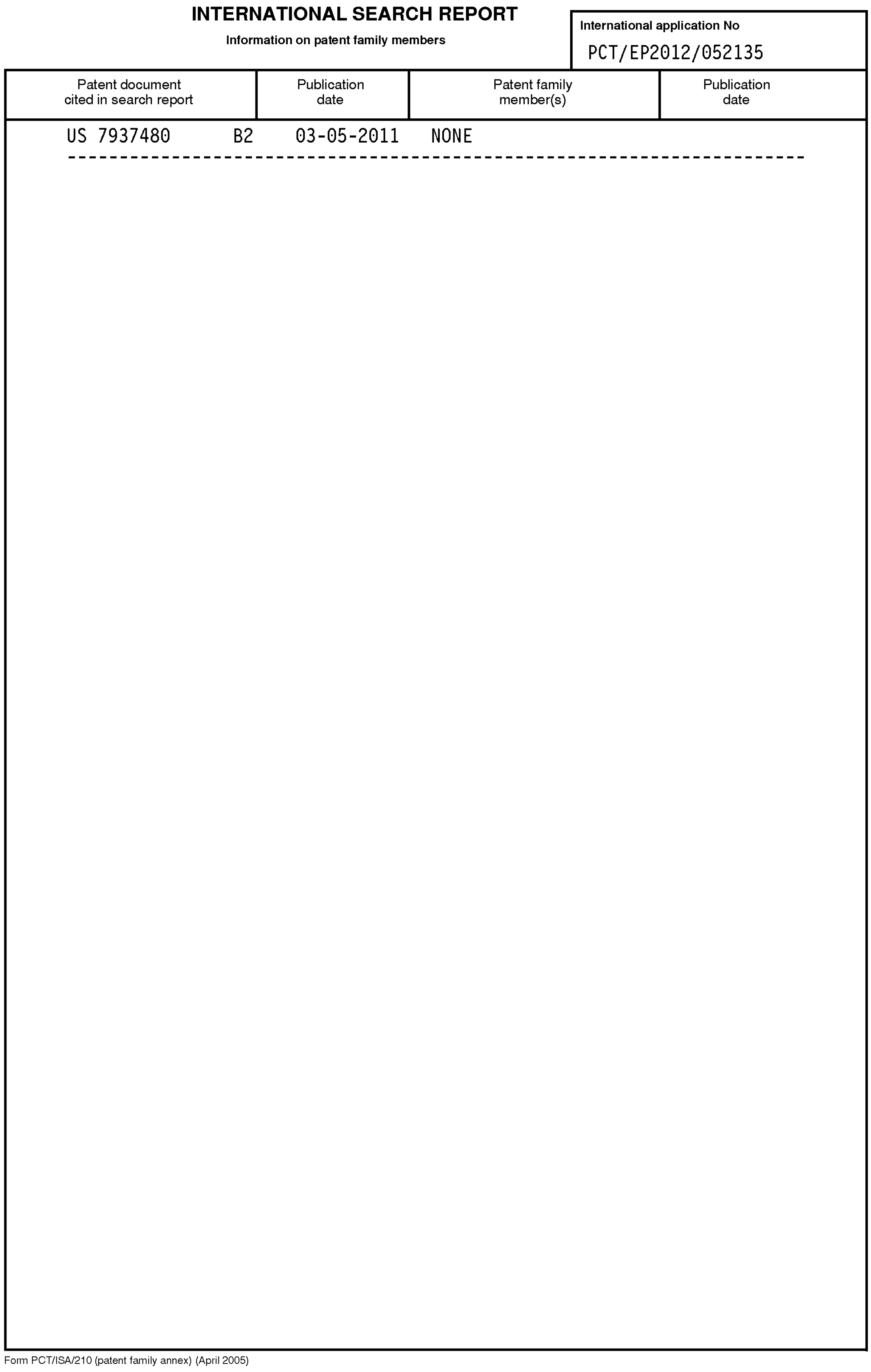
INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2012/052135

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	ISAAC AGUDO ET AL: "A Multidimensional Reputation Scheme for Identity Federations", 10 September 2009 (2009-09-10), PUBLIC KEY INFRASTRUCTURES, SERVICES AND APPLICATIONS, SPRINGER BERLIN HEIDELBERG, BERLIN, HEIDELBERG, PAGE(S) 225 - 238, XP019154564, ISBN: 978-3-642-16440-8	1-7,9-19
A	abstract 1. Introduction 4. Federated Reputation System -----	8,20,21
X	SABATER-MIR ET AL: "On representation and aggregation of social evaluations in computational trust and reputation models", INTERNATIONAL JOURNAL OF APPROXIMATE REASONING, ELSEVIER SCIENCE, NEW YORK, NY, US, vol. 46, no. 3, 15 November 2007 (2007-11-15), pages 458-483, XP022346007, ISSN: 0888-613X, DOI: 10.1016/J.IJAR.2006.12.013	1-7,9-19
A	the whole document -----	8,20,21
A	PHILLIP J. WINDLEY ET AL: "Using reputation to augment explicit authorization", PROCEEDINGS OF THE 2007 ACM WORKSHOP ON DIGITAL IDENTITY MANAGEMENT , DIM '07, 1 January 2007 (2007-01-01), page 72, XP55037844, New York, New York, USA DOI: 10.1145/1314403.1314416 ISBN: 978-1-59-593889-3 cited in the application the whole document -----	1-21
A	FÉLIX GÓMEZ MÁRMOL ET AL: "Towards pre-standardization of trust and reputation models for distributed and heterogeneous systems", COMPUTER STANDARDS & INTERFACES, vol. 32, no. 4, 1 June 2010 (2010-06-01), pages 185-196, XP55037846, ISSN: 0920-5489, DOI: 10.1016/j.csi.2010.01.003 cited in the application the whole document -----	1-21

Form PCT/ISA/210 (continuation of second sheet) (April 2005)



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/EP2012/052135

Patent document cited in search report		Publication date	Patent family member(s)	Publication date
US 7937480	B2	03-05-2011	NONE	

Form PCT/ISA/210 (patent family annex) (April 2005)

A.II Method to support an advanced home services coordination platform

Title:	Method to support an advanced home services coordination platform
Authors:	Joao Girao, Brigitta Lange, Nils Gruschka, Ginés Dólera Tormo, Félix Gómez Mármol
Type:	Unites States Patent Application Publication
Publication Number:	US 2013/0304488 A1
Year:	2013
Month:	November
Day:	14
State:	Published

US 20130304488A1

(19) **United States**

(12) **Patent Application Publication**

Girao et al.

(10) **Pub. No.: US 2013/0304488 A1**

(43) **Pub. Date: Nov. 14, 2013**

(54) **METHOD TO SUPPORT AN ADVANCED HOME SERVICES COORDINATION PLATFORM**

(71) Applicant: **NEC EUROPE LIMITED**, London (GB)

(72) Inventors: **Joao Girao**, Ludwigshafen (DE); **Brigitta Lange**, Vaihingen/Enz (DE); **Nils Gruschka**, Hemmingstedt (DE); **Gines Dolera Tormo**, Heidelberg (DE); **Felix Gomez Marmol**, Heidelberg (DE)

(21) Appl. No.: **13/889,613**

(22) Filed: **May 8, 2013**

Related U.S. Application Data

(60) Provisional application No. 61/644,501, filed on May 9, 2012.

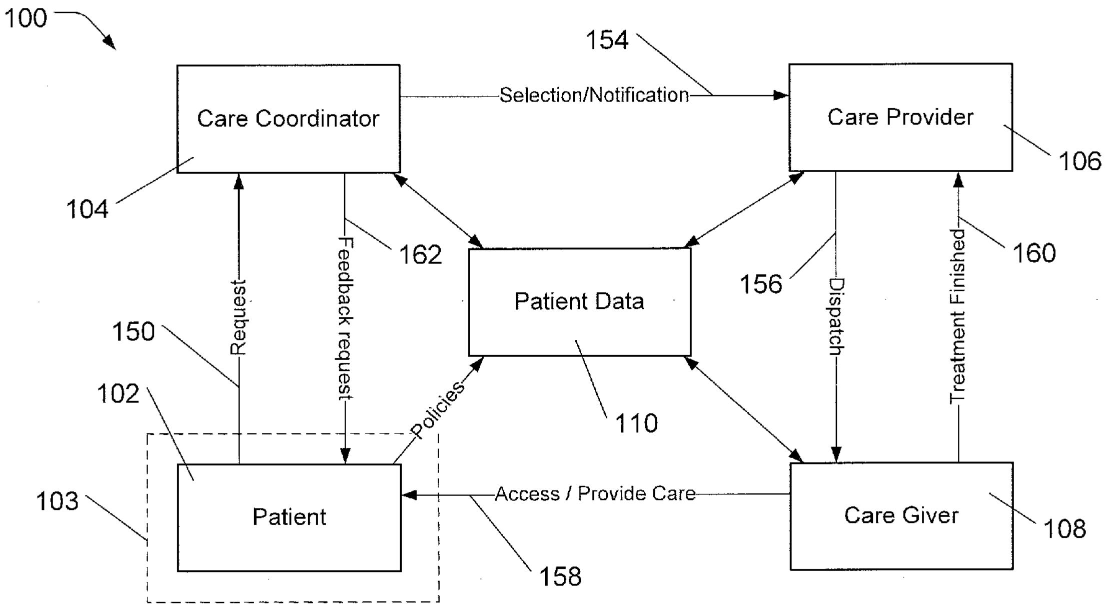
Publication Classification

(51) **Int. Cl.**
G06F 19/00 (2006.01)

(52) **U.S. Cl.**
CPC *G06F 19/3418* (2013.01)
USPC **705/2**

(57) **ABSTRACT**

A method coordinating home services is provided, including receiving a request for home services from a customer over a network and forwarding the request from the customer to a home services coordinator over the network. A reputation system assists the home services coordinator to select a service provider based on customer needs, preferences, and a reputation of the service provider. Information sufficient to permit the service provider to select a home delivery provider that can satisfy customer needs is provided to the selected service provider over the network. The selected home delivery provider is provided with access to customer data and with access to a customer physical system over the network, to provide the service. Feedback is requested from the customer after the service has been delivered, and is used in the reputation system to update the customer preferences and the reputation of the service provider.



Patent Application Publication Nov. 14, 2013 Sheet 1 of 4 US 2013/0304488 A1

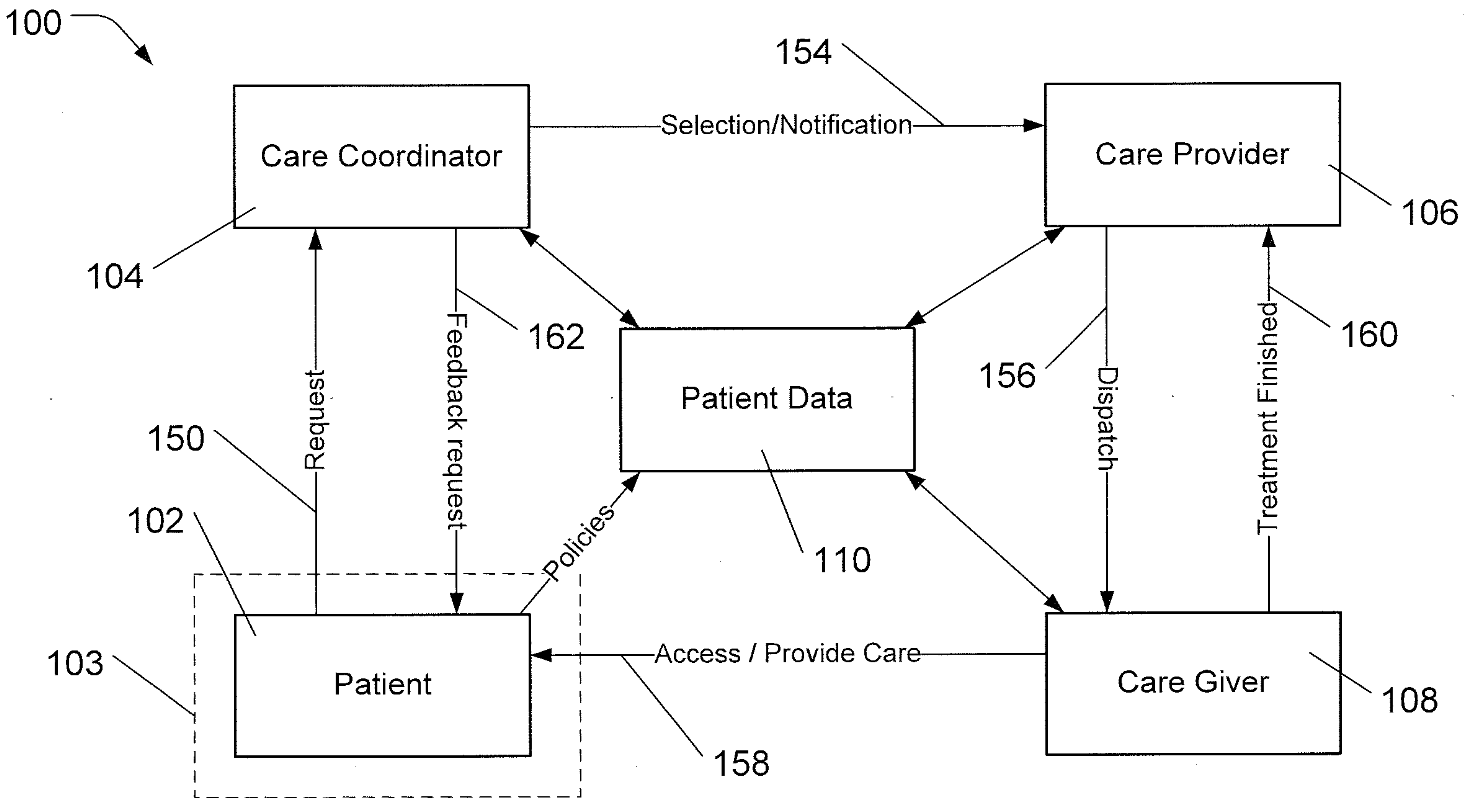


FIG. 1

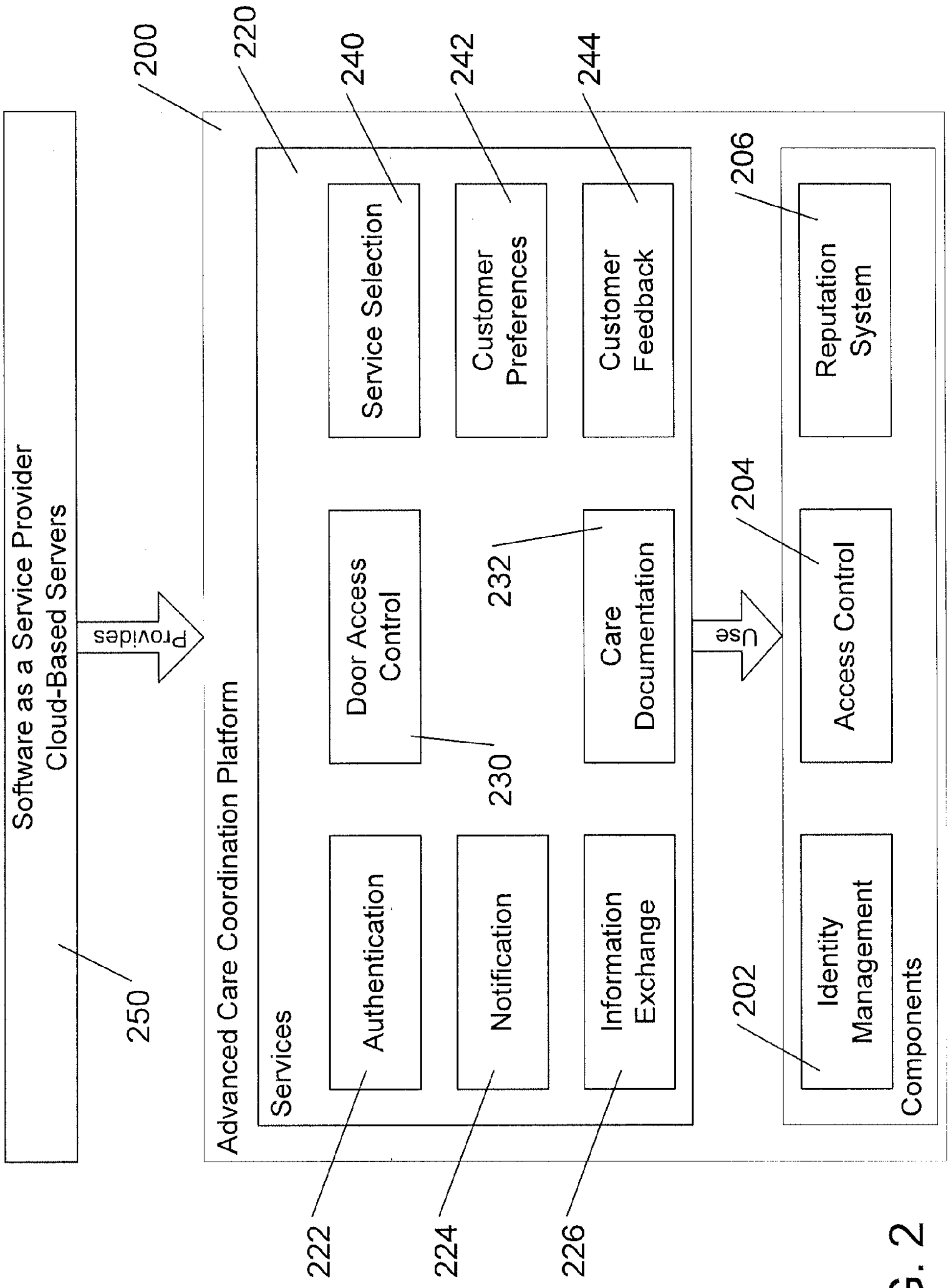

Software as a Service Provider
Cloud-Based Servers
250
Provides
200
Advanced Care Coordination Platform
220
Services
Authentication
222
Notification
224
Information Exchange
226
Door Access Control
230
Care Documentation
232
Service Selection
240
Customer Preferences
242
Customer Feedback
244
Use
Components
Identity Management
202
Access Control
204
Reputation System
206

FIG. 2

Patent Application Publication Nov. 14, 2013 Sheet 3 of 4 US 2013/0304488 A1

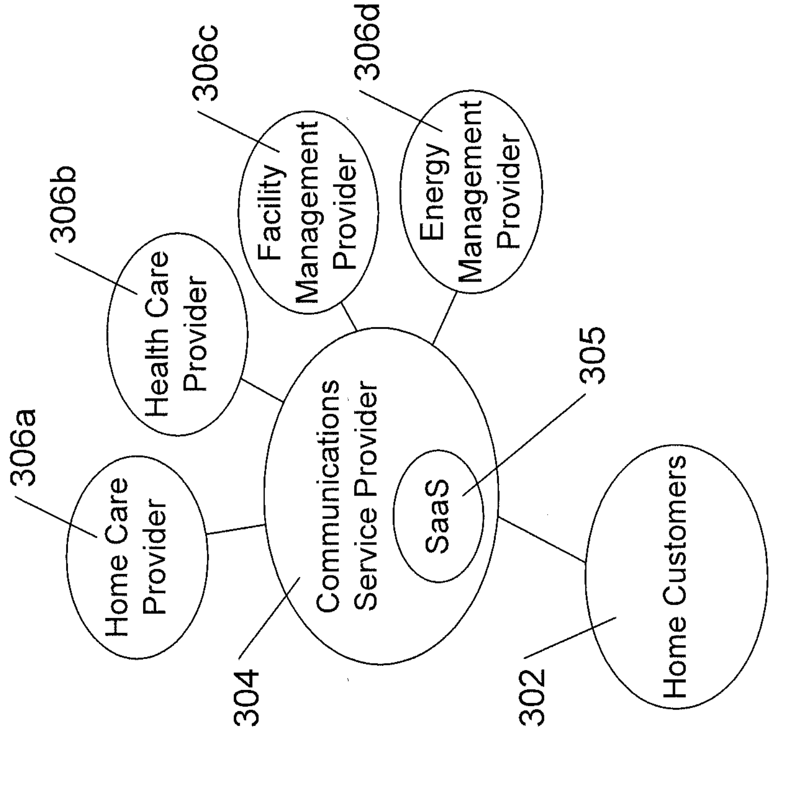

306a
Home Care Provider
306b
Health Care Provider
306c
Facility Management Provider
306d
Energy Management Provider
Communications Service Provider
SaaS
305
304
302
Home Customers

FIG. 3A

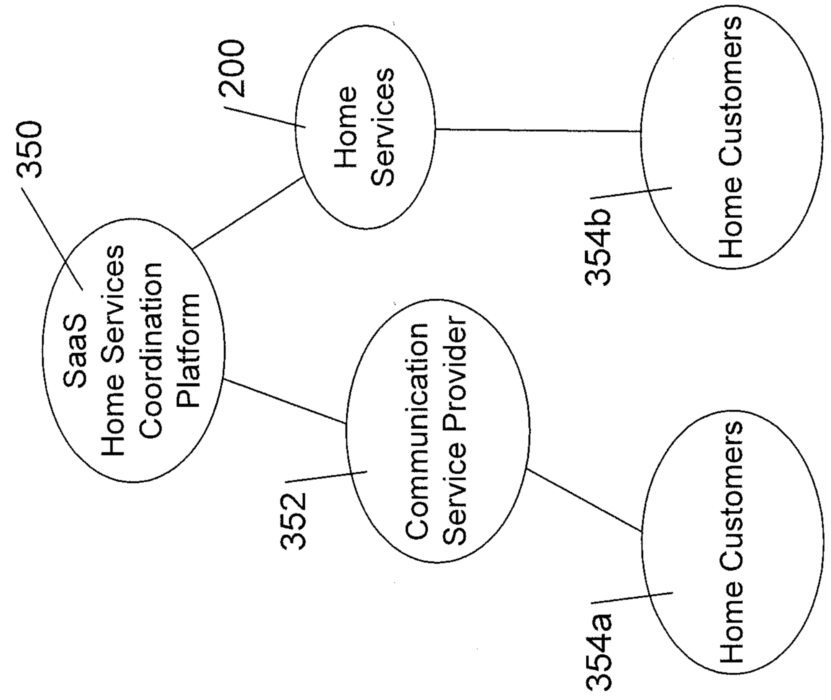

350
SaaS Home Services Coordination Platform
200
Home Services
352
Communication Service Provider
354a
Home Customers
354b
Home Customers

FIG. 3B

Patent Application Publication **Nov. 14, 2013 Sheet 4 of 4** **US 2013/0304488 A1**

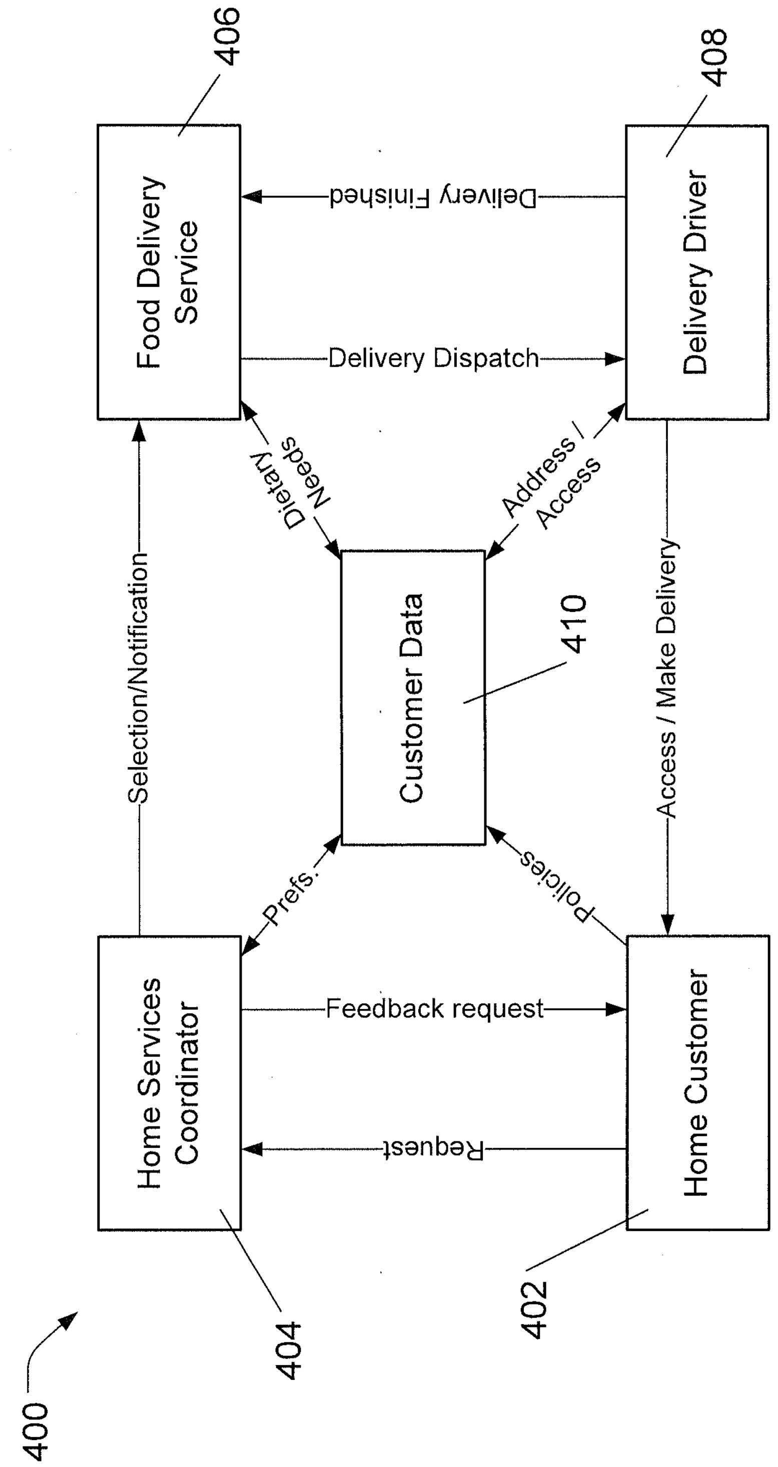
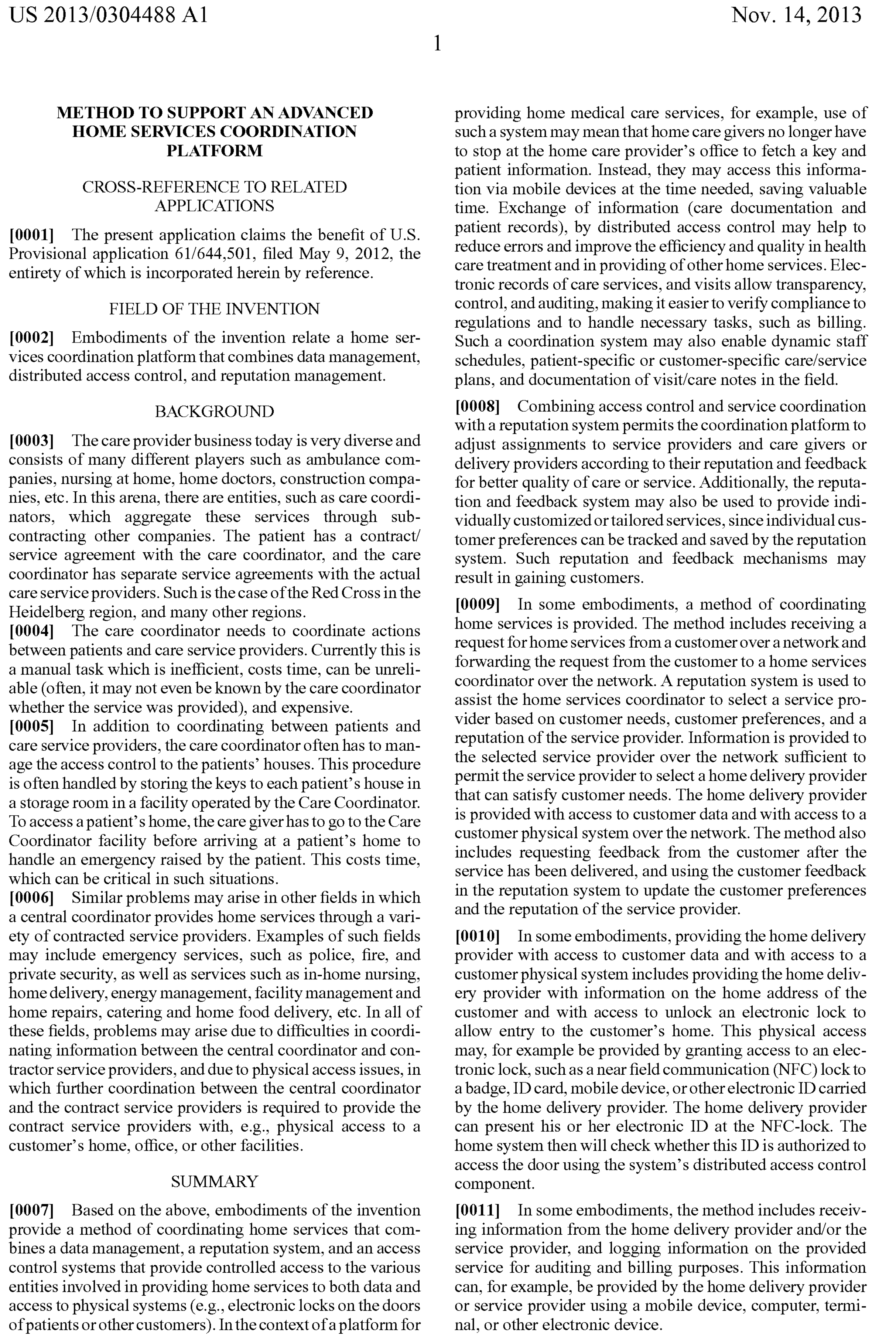


FIG. 4



METHOD TO SUPPORT AN ADVANCED HOME SERVICES COORDINATION PLATFORM

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] The present application claims the benefit of U.S. Provisional application 61/644,501, filed May 9, 2012, the entirety of which is incorporated herein by reference.

FIELD OF THE INVENTION

[0002] Embodiments of the invention relate a home services coordination platform that combines data management, distributed access control, and reputation management.

BACKGROUND

[0003] The care provider business today is very diverse and consists of many different players such as ambulance companies, nursing at home, home doctors, construction companies, etc. In this arena, there are entities, such as care coordinators, which aggregate these services through subcontracting other companies. The patient has a contract/service agreement with the care coordinator, and the care coordinator has separate service agreements with the actual care service providers. Such is the case of the Red Cross in the Heidelberg region, and many other regions.

[0004] The care coordinator needs to coordinate actions between patients and care service providers. Currently this is a manual task which is inefficient, costs time, can be unreliable (often, it may not even be known by the care coordinator whether the service was provided), and expensive.

[0005] In addition to coordinating between patients and care service providers, the care coordinator often has to manage the access control to the patients' houses. This procedure is often handled by storing the keys to each patient's house in a storage room in a facility operated by the Care Coordinator. To access a patient's home, the care giver has to go to the Care Coordinator facility before arriving at a patient's home to handle an emergency raised by the patient. This costs time, which can be critical in such situations.

[0006] Similar problems may arise in other fields in which a central coordinator provides home services through a variety of contracted service providers. Examples of such fields may include emergency services, such as police, fire, and private security, as well as services such as in-home nursing, home delivery, energy management, facility management and home repairs, catering and home food delivery, etc. In all of these fields, problems may arise due to difficulties in coordinating information between the central coordinator and contractor service providers, and due to physical access issues, in which further coordination between the central coordinator and the contract service providers is required to provide the contract service providers with, e.g., physical access to a customer's home, office, or other facilities.

SUMMARY

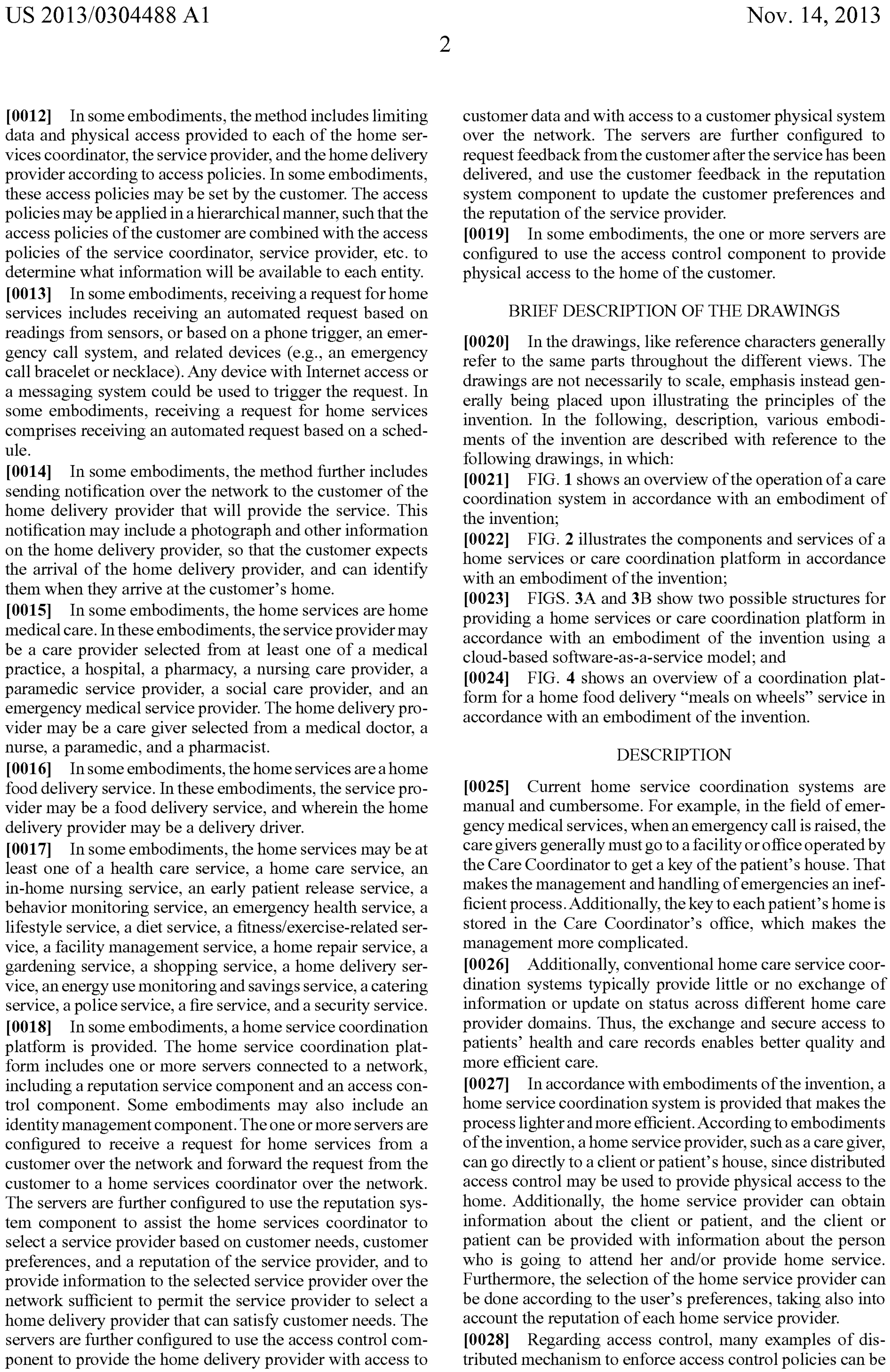
[0007] Based on the above, embodiments of the invention provide a method of coordinating home services that combines a data management, a reputation system, and an access control systems that provide controlled access to the various entities involved in providing home services to both data and access to physical systems (e.g., electronic locks on the doors of patients or other customers). In the context of a platform for providing home medical care services, for example, use of such a system may mean that home care givers no longer have to stop at the home care provider's office to fetch a key and patient information. Instead, they may access this information via mobile devices at the time needed, saving valuable time. Exchange of information (care documentation and patient records), by distributed access control may help to reduce errors and improve the efficiency and quality in health care treatment and in providing of other home services. Electronic records of care services, and visits allow transparency, control, and auditing, making it easier to verify compliance to regulations and to handle necessary tasks, such as billing. Such a coordination system may also enable dynamic staff schedules, patient-specific or customer-specific care/service plans, and documentation of visit/care notes in the field.

[0008] Combining access control and service coordination with a reputation system permits the coordination platform to adjust assignments to service providers and care givers or delivery providers according to their reputation and feedback for better quality of care or service. Additionally, the reputation and feedback system may also be used to provide individually customized or tailored services, since individual customer preferences can be tracked and saved by the reputation system. Such reputation and feedback mechanisms may result in gaining customers.

[0009] In some embodiments, a method of coordinating home services is provided. The method includes receiving a request for home services from a customer over a network and forwarding the request from the customer to a home services coordinator over the network. A reputation system is used to assist the home services coordinator to select a service provider based on customer needs, customer preferences, and a reputation of the service provider. Information is provided to the selected service provider over the network sufficient to permit the service provider to select a home delivery provider that can satisfy customer needs. The home delivery provider is provided with access to customer data and with access to a customer physical system over the network. The method also includes requesting feedback from the customer after the service has been delivered, and using the customer feedback in the reputation system to update the customer preferences and the reputation of the service provider.

[0010] In some embodiments, providing the home delivery provider with access to customer data and with access to a customer physical system includes providing the home delivery provider with information on the home address of the customer and with access to unlock an electronic lock to allow entry to the customer's home. This physical access may, for example be provided by granting access to an electronic lock, such as a near field communication (NFC) lock to a badge, ID card, mobile device, or other electronic ID carried by the home delivery provider. The home delivery provider can present his or her electronic ID at the NFC-lock. The home system then will check whether this ID is authorized to access the door using the system's distributed access control component.

[0011] In some embodiments, the method includes receiving information from the home delivery provider and/or the service provider, and logging information on the provided service for auditing and billing purposes. This information can, for example, be provided by the home delivery provider or service provider using a mobile device, computer, terminal, or other electronic device.



[0012] In some embodiments, the method includes limiting data and physical access provided to each of the home services coordinator, the service provider, and the home delivery provider according to access policies. In some embodiments, these access policies may be set by the customer. The access policies may be applied in a hierarchical manner, such that the access policies of the customer are combined with the access policies of the service coordinator, service provider, etc. to determine what information will be available to each entity.

[0013] In some embodiments, receiving a request for home services includes receiving an automated request based on readings from sensors, or based on a phone trigger, an emergency call system, and related devices (e.g., an emergency call bracelet or necklace). Any device with Internet access or a messaging system could be used to trigger the request. In some embodiments, receiving a request for home services comprises receiving an automated request based on a schedule.

[0014] In some embodiments, the method further includes sending notification over the network to the customer of the home delivery provider that will provide the service. This notification may include a photograph and other information on the home delivery provider, so that the customer expects the arrival of the home delivery provider, and can identify them when they arrive at the customer's home.

[0015] In some embodiments, the home services are home medical care. In these embodiments, the service provider may be a care provider selected from at least one of a medical practice, a hospital, a pharmacy, a nursing care provider, a paramedic service provider, a social care provider, and an emergency medical service provider. The home delivery provider may be a care giver selected from a medical doctor, a nurse, a paramedic, and a pharmacist.

[0016] In some embodiments, the home services are a home food delivery service. In these embodiments, the service provider may be a food delivery service, and wherein the home delivery provider may be a delivery driver.

[0017] In some embodiments, the home services may be at least one of a health care service, a home care service, an in-home nursing service, an early patient release service, a behavior monitoring service, an emergency health service, a lifestyle service, a diet service, a fitness/exercise-related service, a facility management service, a home repair service, a gardening service, a shopping service, a home delivery service, an energy use monitoring and savings service, a catering service, a police service, a fire service, and a security service.

[0018] In some embodiments, a home service coordination platform is provided. The home service coordination platform includes one or more servers connected to a network, including a reputation service component and an access control component. Some embodiments may also include an identity management component. The one or more servers are configured to receive a request for home services from a customer over the network and forward the request from the customer to a home services coordinator over the network. The servers are further configured to use the reputation system component to assist the home services coordinator to select a service provider based on customer needs, customer preferences, and a reputation of the service provider, and to provide information to the selected service provider over the network sufficient to permit the service provider to select a home delivery provider that can satisfy customer needs. The servers are further configured to use the access control component to provide the home delivery provider with access to customer data and with access to a customer physical system over the network. The servers are further configured to request feedback from the customer after the service has been delivered, and use the customer feedback in the reputation system component to update the customer preferences and the reputation of the service provider.

[0019] In some embodiments, the one or more servers are configured to use the access control component to provide physical access to the home of the customer.

BRIEF DESCRIPTION OF THE DRAWINGS

[0020] In the drawings, like reference characters generally refer to the same parts throughout the different views. The drawings are not necessarily to scale, emphasis instead generally being placed upon illustrating the principles of the invention. In the following, description, various embodiments of the invention are described with reference to the following drawings, in which:

[0021] FIG. **1** shows an overview of the operation of a care coordination system in accordance with an embodiment of the invention;

[0022] FIG. **2** illustrates the components and services of a home services or care coordination platform in accordance with an embodiment of the invention;

[0023] FIGS. **3**A and **3**B show two possible structures for providing a home services or care coordination platform in accordance with an embodiment of the invention using a cloud-based software-as-a-service model; and

[0024] FIG. **4** shows an overview of a coordination platform for a home food delivery "meals on wheels" service in accordance with an embodiment of the invention.

DESCRIPTION

[0025] Current home service coordination systems are manual and cumbersome. For example, in the field of emergency medical services, when an emergency call is raised, the care givers generally must go to a facility or office operated by the Care Coordinator to get a key of the patient's house. That makes the management and handling of emergencies an inefficient process. Additionally, the key to each patient's home is stored in the Care Coordinator's office, which makes the management more complicated.

[0026] Additionally, conventional home care service coordination systems typically provide little or no exchange of information or update on status across different home care provider domains. Thus, the exchange and secure access to patients' health and care records enables better quality and more efficient care.

[0027] In accordance with embodiments of the invention, a home service coordination system is provided that makes the process lighter and more efficient. According to embodiments of the invention, a home service provider, such as a care giver, can go directly to a client or patient's house, since distributed access control may be used to provide physical access to the home. Additionally, the home service provider can obtain information about the client or patient, and the client or patient can be provided with information about the person who is going to attend her and/or provide home service. Furthermore, the selection of the home service provider can be done according to the user's preferences, taking also into account the reputation of each home service provider.

[0028] Regarding access control, many examples of distributed mechanism to enforce access control policies can be found in the literature. However, they are generally related to accessing information or data. Physical access is generally managed locally, where actors involved are known beforehand, and local policies can be directly established. In other words, in this environment, a system administrator is in charge of managing these policies, granting or denying access to specific individuals during a specific periods of time. According to embodiments of the invention, the physical access is instead managed in a distributed way. This permits, for example, a care giver to access a patient's house just when an emergency occurs, although the patient does not necessarily know the identity of the care giver beforehand.

[0029] While the examples below will most often be discussed in the context of providing home medical care, particularly on an emergency basis, it will be understood that similar systems and methods could be applied to a wide range of home service or home care scenarios. For example, similar systems and methods could be applied to fields such as emergency services, such as police, fire, and private security, as well as services such as in-home nursing, home delivery, energy management, facility management and home repairs, catering and home food delivery, etc.

[0030] Referring now to FIG. **1**, an overview of the operation of a care coordination system **100** in accordance with an embodiment of the invention is described. The system involves a number of entities, including a patient **102**, who is generally located in his or her home **103**. The patient **102** may be, for example, a person who requires home care and uses or depends on ambient assisted living (AAL) applications in his/her home environment (e.g., an elderly person). The patient **102** subscribes to an emergency medical service offered through a care coordinator **104**.

[0031] The care coordinator **104** provides and maintains a home emergency call/telemonitoring/telecare infrastructure. Generally, the care coordinator **104** has a call center and coordinates help and home care services (e.g., medical doctors, emergency vehicles, nursing services, etc.). The care coordinator **104** maintains relationships with numerous care providers **106**.

[0032] The care providers **106** provide home care services in the geographical area of the patient **102** (but do not themselves need to be located in the geographical area of the patient **102**), and coordinate mobile care givers. In the context of providing medical care, the care providers may include entities such as hospitals, medical practices, pharmacies, nursing care providers, physiotherapy providers, and so on, providing home care, social care, nursing services, physiotherapy, and home delivery of pharmaceuticals. The care providers **106** maintain relationships with numerous mobile/home care givers **108**.

[0033] The mobile care givers are prepared to provide care in the home of the patient **102**. The mobile care givers may be, e.g. individual doctors, nurses, paramedics, physiotherapists, etc. who provide home care services at the patient's home, and have mobile devices to communicate with the coordinator.

[0034] Patient data **110** is also important in the system according to various embodiments of the invention. Patient data **110** is handled in a distributed manner in the system, so that the patient **102** ultimately controls the rules for access to the data, and so that each of the other entities—the care coordinator **104**, the care providers **106**, and the home care givers **108** each receive the information that they need to provide their respective services. It should be understood that although the patient data **110** is shown in FIG. **1** as a single item, in fact, the patient data **110** may be distributed, and may be stored in a variety of places. Additionally, it will be understood that in some embodiments there may be access rules that are not controlled by the patient, such as rules that provide access and exchange of patient data for legal or other requirements. For example, in case of an emergency, an emergency doctor may be permitted to access relevant patient data stored and controlled by the care coordinator. Of course, all of the various stake holders, including the patient **102**, care coordinator **104**, care providers **106**, and care giver **108** can access or exchange a variety of data other than the patient data **110**, and/or not necessarily governed by the patient **102**, such as care documentation for billing, accounting, and/or insurance purposes.

[0035] FIG. **1** also shows communications in an example scenario for use of the system **100**, in accordance with an embodiment of the invention. To start the scenario, a patient **102** has an emergency or a scheduled house call. At **150**, the care coordinator **104** is alerted that a service is required by the patient **102**. This alert can be based on sensors in the home **103** of the patient **102**, on pattern recognition, on an explicit trigger by the patient (e.g., the patient presses a medical panic or emergency button), or on a scheduled event (e.g., as part of a treatment).

[0036] The care coordinator **104** selects a care provider **106** based on the needs of the patient. This involves determining both the type or types of care provider that is to be selected (e.g., nurse providers), and the specific provider or providers of that type that will provide the service. This selection may also be made using the reputation management capabilities of the system. This permits a care provider **106** to be selected based in part on the previous patient experience with the available care providers **106**. The final selection can be made automatically based on the previous experience and the scoring of the care providers **106** by the patient **102**. Alternatively, the selection of the care provider **106** may be made manually by the care coordinator **104**, using, e.g., scores that rate different care providers **106** of the type needed by the patient **102**. Once a care provider **106** is selected, at **154**, the care coordinator **104** informs the selected care provider **106** that a new call exists.

[0037] At **156**, the care provider allocates one or more care givers **108**, such as nurses, doctors, drivers, etc., and dispatches them. It is assumed in this scenario that the selected care givers **108** are already authenticated with the care coordination platform through the care provider **106**. At **158**, the care givers **108** are given access to all required systems through the care provider **106**, the care coordinator **104**, and/or the patient **102**. These systems may include access to data, such as patient records held by the care coordinator **104** or by other institutions, and access to physical systems, such as access to the home **103** of the patient **102**, through, e.g., a near-field communication (NFC) based door lock that can be unlocked using a care giver's ID card, badge, mobile device, or other electronic or RFID-based ID. Access to these systems can be appropriately logged, for security and for billing purposes. Additionally, as will be discussed below, the care givers **108** will no longer have access to the door after they are finished providing the required care to the patient **102**.

[0038] In accordance with some embodiments, when access to a system is required, the request is first made to the care coordinator **104**, who will identify the care provider **106** and combine its decision with respect to access with the decision response from the care provider **106**. When receiving the policy decision request, the care provider **106** will identify the subject (e.g., the care giver **108**) and confirm the access permission with its policies.

[0039] Additionally, in some embodiments, once the care givers **108** are allocated by the care provider **106**, the patient **102** will receive a message from the system **100** informing him or her of the care givers **108** that have been assigned, and of their expected arrival. In some of these embodiments, the message may include, e.g., photographs of the care givers **108**, so they can be recognized by the patient **102** on their arrival.

[0040] At **160**, once the service has been provided, the care givers **108** will signal that the treatment is finished, for example using their mobile devices or other electronic devices. The care givers **108** will lose all access to the systems relates to the treatment, including, for example, physical access (e.g., to the patient's home) and data access (e.g., to the patient's data). Details on the treatment (including, e.g., a time stamp) will be recorded by the system in order to allow later auditing, which may be required in some jurisdictions, and for possible use in billing for the treatments and services that were provided.

[0041] At **162**, the patient **102** will receive a feedback/quality control request to rate the treatment that was just received, for use in the reputation management system. This feedback could be given, for example, by filling in a physical or an electronic form (e.g., via the Internet or Web), verbally, or in natural language. The feedback is then added to the rest of the feedback related to the care provider **106**. In some embodiments, a score may be calculated and a new assessment of the care provider **106** will be added to the profile of the care provider **106**, which may be used by the care coordinator **104** in selecting care providers.

[0042] The system **100** respects the privacy of the patient **102**. For example, the care provider **106** does not need to know who the patient is, only the care coordinator **104** and care givers **108** need to know, and the care coordinator **104** does not need to directly know which care givers **108** were involved in the treatment. An identity management component, as will be described below, can be used to assign virtual identities (pseudonyms). The access control policies are distributed, and could be enforced both locally and remotely by the entities with the right data. The system **100** may interface with both physical and online systems, making use of patient feedback for automated care provider selection. The system **100** manages the physical access control in a distributed way, so patients can temporally allow access to the care givers although they do not know the identity of the care giver that will attend to their emergency home care.

[0043] For example, an access control policy set by the patient **102** may be: "allow access to the door to Bob, Carol, and to people sent by the care coordinator". This policy selects particular people to have access to the door (Bob and Carol), and delegates the access decision to the care coordinator **104**. The care coordinator **104** may have an access control policy for the patients door and patient information as follows: "permit access if an alert in the patient's house is still active and the care giver has been sent by the care provider." Thus, the care coordinator **104** does not need to manage the care coordination information of the care givers **108**, providing increased privacy. The care provider **106** may have an access policy such as "Alice is attending the emergency xx:yy." This policy provides access to a particular care giver **108**, assuming that the care provider **106** has the ability to provide such access. The care provider **106** generally cannot manage patient information without consent.

[0044] Referring now to FIG. **2**, the basic structure of the system is described. The advanced care coordination platform **200** includes three main components—an identity management component **202**, an access control component **204**, and a reputation system **206**. The identity management component **202** supports complex ID brokerage scenarios, cross protocol single sign on, and manages authentication and virtual identities for privacy. The identity management component **202** provides services **220** in the system such as authentication services **222** for users of the system (patients, care coordinators, care providers, care givers), notification services **224** (e.g., directing messages to the proper individuals whose identities have been properly authenticated), and information exchange services **226**, such as providing access to distributed electronic health records (EHR).

[0045] The access control component **204** provides transparency and confidentiality by supporting hierarchical access requests, and dynamic references to other authoritative domains. Services such as the door access control service **230**, which controls physical access to patients' homes, make use of the access control component **204** and the identity management component **202**. In some embodiments, similar services (not shown) may use the access control component **204** to access a variety of other physical devices associated with a customer or patient, such as security cameras, sensor readings, control of appliances, heating, lighting, etc., depending on the field in which the system is being used. Other services, such as the care documentation service **232**, or other services (not shown) that access patient data, including, e.g., the information exchange services **226** described above, may also make use of the access control component **204** to control access to private data.

[0046] In some embodiments, the policies used by the access control component **204** are stored locally by each entity that is part of the system, and may be evaluated locally, but in a hierarchical manner. The policies that are used to control access in the access control component **204** may be evaluated only when needed in some embodiments. Thus, each patient can set his or her own policies for access to physical systems such as his or her door, and can specify who or what class of people may have access. For example, a patient could specify that only particular care providers get access, or that all doctors can have access, or that only specific people get access, or any policy that the patient wishes to establish. These policies (or the results of evaluating these policies as needed) are handled in a hierarchical manner with those of other entities in the system. For example, the policies of the patient may be combined with access policies associated with the care coordinator with which the patient has subscribed, and the care provider that has been selected to provide care to determine what kind of access to data or physical systems will be granted to a particular care giver. In some embodiments, certain policies may be mandatory, such as when access policies are set to comply with local data protection or privacy laws.

[0047] The reputation system component **206** supports customized and reputation-based service selection. For example, the reputation system component may manage the reputations of care providers, and update reputation information according to a customer or patient feedback mechanism. Services such as a service selection service **240**, which, e.g.,

assists the care coordinator in selecting a care provider, may use the reputation system component **206**. Other services, such as the customer preferences service **242**, which permits a customer or patient to provide his or her preferences to the system, and a customer feedback service **244**, which obtains feedback information from customers or patients, also use the reputation system component **206**.

[0048] In some embodiments, further components (not shown) may be included in the advanced care coordination platform **200**, such as a components for managing sensors and other components of an ambient assisted living (AAL) system or automated home, and for offering a variety of AAL-related services.

[0049] In some embodiments, the components and services described above may be offered through a cloud-based software-as-a-service (SaaS) model. In such a model, the various components and services are offered over the Internet, and may be operated on one or more server computers (e.g., having a processor, memory, storage, etc.) connected to the Internet, and operated by a SaaS provider **250**. By using such a SaaS model, the care coordinator and the care providers can run the system with only limited IT infrastructure. Generally, the care coordinator, care givers and patients may need only Web browser access, and in some embodiments may interact with the system over the Web.

[0050] Two alternative setups of the system on a SaaS model are shown in FIGS. **3**A and **3**B. In FIG. **3**A, home customers **302** connect to a communication service provider **304**, such as a telephone company, cable company, Internet provider, etc., that includes an advanced home services coordination platform **305** in accordance with embodiments of the invention. Various home service providers **306***a*-**306***d*, such as home care provider **306***a*, health care provider **306***b*, facility management provider **306***c* and energy management provider **306***d* are also connected to the communication service provider **304**, through which they are connected to and participate in the advanced home services coordination platform **304**. The communication service provider operates the home services coordination platform **304**, and accepts subscriptions to the platform from the home customers **302** and from the various home service providers **306***a*-**306***d*, as separate paid subscriptions, or as part of their basic communications subscription or service, or as part of an add-on package subscription or service.

[0051] In FIG. **3**B, the advanced home services coordination platform **350** operates as a separate service, communicating with, e.g., 3rd parry service providers, such as communication or Internet service providers **352**, which are connected to a first set of home customers **354***a*. The advanced home services coordination platform **350** also communicates with a set of home services **356**, which may be directly connected to a second set of home customers **354***b*. Both the first and second sets of home customers **354***a* and **354***b* may take advantage of the services offered through their service providers, which in turn use the advanced home services coordination platform **350** to offer these services. Other service providers (not shown), with their own sets of home subscribers (not shown) could also connect to the cloud-based advanced home services coordination platform **350**, to take advantage of its coordination capabilities, as described above, while offering differing sets of services at varying pricing to their home customers.

[0052] FIG. **4** shows an example of a system **400**, in which an advanced home services coordination platform according to various embodiments of the invention is used for a home delivery "meals-on-wheels" service. In this system, which operates in a manner similar to the emergency health-care systems described above, home customers **402** request meal delivery service through a home services coordinator **404**. This service request may occur manually, or automatically, e.g., according to a schedule. The home services coordinator **404** then uses data on the personal preferences of the home customer **402** from customer data **410** and the reputation scores of food delivery services **406** to select a food delivery service **406** to handle the order. When it receives the order through the home services coordinator **404**, the food delivery service **406** may access preferences, dietary requirements, etc. for the home customer **402**. In some embodiments, this may involve accessing sensor data, such as the current blood sugar level of the home customer **402**. This information is used by the food delivery service **406** to select a menu that will meet the needs of the home customer **402**. The food delivery service **406** then prepares (or otherwise obtains) the food to be sent to the home customer **402**, and dispatches a delivery driver **408**. The delivery driver **408** is provided with information on the home customer's address, and may be given access to open the door of the home customer **402** in order to deliver the prepared meal. After the meal is delivered, the delivery driver **408** logs this information for auditing and billing purposes. The system also ends any access that the delivery driver **408** may have to the home or door of the home customer **402**, and ends access to any data for the completed delivery. Finally, a feedback request is sent to the home customer **402**, who may provide feedback on the service and quality that may be used to adjust the home customer's preferences and the reputation scores of the food delivery service **406**, which will be used by the home services coordinator **404** in future selection of food delivery services.

[0053] While FIG. **4** shows use of a system for a "meals-on-wheels" service, it will be understood that many other possible services could use the same advanced home services platform. As seen above, such a platform could be used to provide healthcare services, such as home care, early patient release, behavior monitoring, and emergency health services. A similar platform could be used to offer a variety of lifestyle services, such as diet and fitness/exercise-related services, facility management services such as home repair or gardening, and other services, such as shopping and home delivery services, energy use monitoring and savings services, security services, and others.

[0054] While the invention has been shown and described with reference to specific embodiments, it should be understood that various changes in form and detail may be made therein without departing from the spirit and scope of the invention as defined by the appended claims. The scope of the invention is thus indicated by the appended claims and all changes which come within the meaning and range of equivalency of the claims are therefore intended to be embraced.

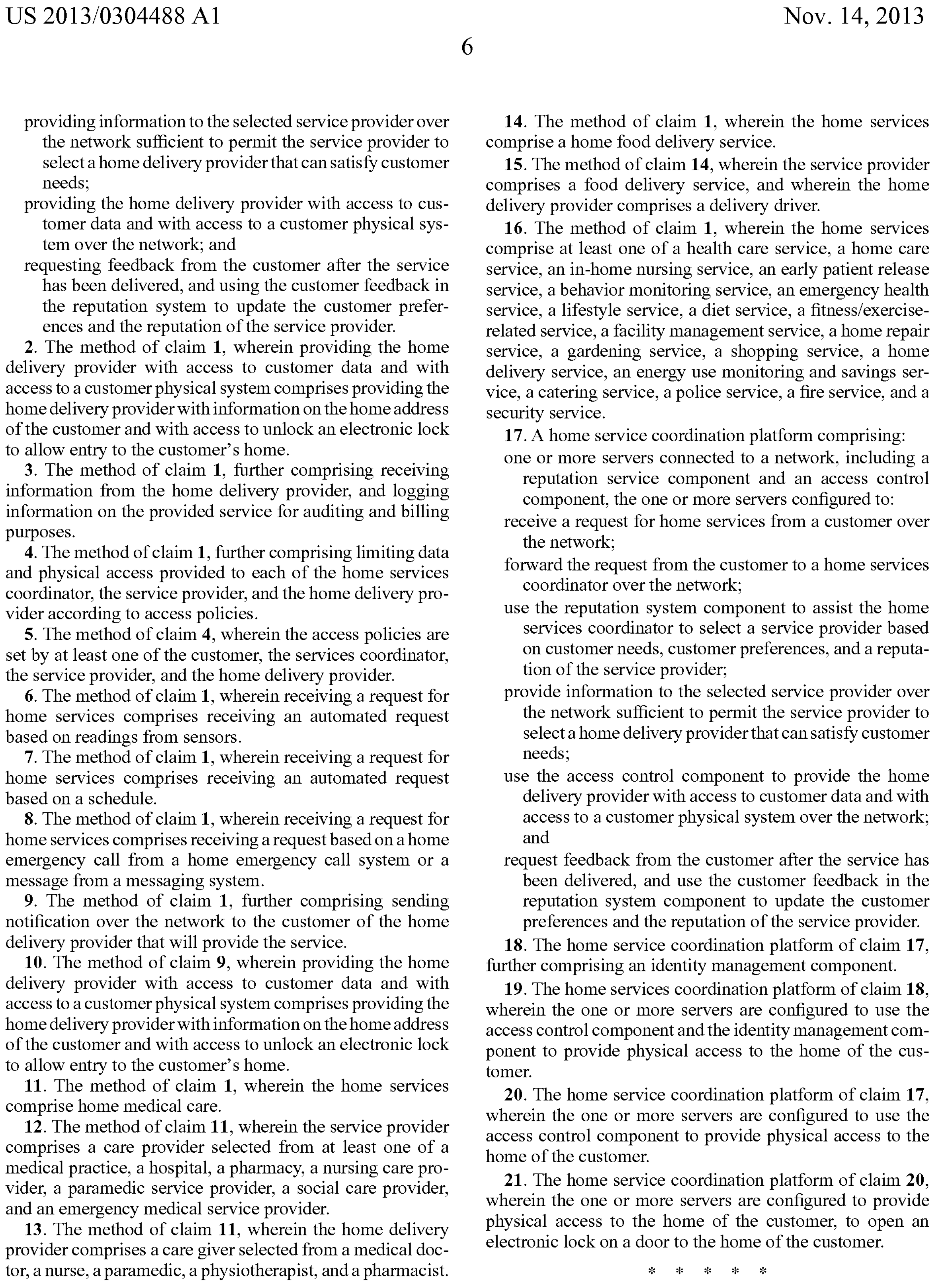
What is claimed is:

1. A method of coordinating home services comprising:

receiving a request for home services from a customer over a network;

forwarding the request from the customer to a home services coordinator over the network;

using a reputation system to assist the home services coordinator to select a service provider based on customer needs, customer preferences, and a reputation of the service provider;



US 2013/0304488 A1 Nov. 14, 2013

6

providing information to the selected service provider over the network sufficient to permit the service provider to select a home delivery provider that can satisfy customer needs;

providing the home delivery provider with access to customer data and with access to a customer physical system over the network; and

requesting feedback from the customer after the service has been delivered, and using the customer feedback in the reputation system to update the customer preferences and the reputation of the service provider.

2. The method of claim **1**, wherein providing the home delivery provider with access to customer data and with access to a customer physical system comprises providing the home delivery provider with information on the home address of the customer and with access to unlock an electronic lock to allow entry to the customer's home.

3. The method of claim **1**, further comprising receiving information from the home delivery provider, and logging information on the provided service for auditing and billing purposes.

4. The method of claim **1**, further comprising limiting data and physical access provided to each of the home services coordinator, the service provider, and the home delivery provider according to access policies.

5. The method of claim **4**, wherein the access policies are set by at least one of the customer, the services coordinator, the service provider, and the home delivery provider.

6. The method of claim **1**, wherein receiving a request for home services comprises receiving an automated request based on readings from sensors.

7. The method of claim **1**, wherein receiving a request for home services comprises receiving an automated request based on a schedule.

8. The method of claim **1**, wherein receiving a request for home services comprises receiving a request based on a home emergency call from a home emergency call system or a message from a messaging system.

9. The method of claim **1**, further comprising sending notification over the network to the customer of the home delivery provider that will provide the service.

10. The method of claim **9**, wherein providing the home delivery provider with access to customer data and with access to a customer physical system comprises providing the home delivery provider with information on the home address of the customer and with access to unlock an electronic lock to allow entry to the customer's home.

11. The method of claim **1**, wherein the home services comprise home medical care.

12. The method of claim **11**, wherein the service provider comprises a care provider selected from at least one of a medical practice, a hospital, a pharmacy, a nursing care provider, a paramedic service provider, a social care provider, and an emergency medical service provider.

13. The method of claim **11**, wherein the home delivery provider comprises a care giver selected from a medical doctor, a nurse, a paramedic, a physiotherapist, and a pharmacist.

14. The method of claim **1**, wherein the home services comprise a home food delivery service.

15. The method of claim **14**, wherein the service provider comprises a food delivery service, and wherein the home delivery provider comprises a delivery driver.

16. The method of claim **1**, wherein the home services comprise at least one of a health care service, a home care service, an in-home nursing service, an early patient release service, a behavior monitoring service, an emergency health service, a lifestyle service, a diet service, a fitness/exercise-related service, a facility management service, a home repair service, a gardening service, a shopping service, a home delivery service, an energy use monitoring and savings service, a catering service, a police service, a fire service, and a security service.

17. A home service coordination platform comprising:

one or more servers connected to a network, including a reputation service component and an access control component, the one or more servers configured to:

receive a request for home services from a customer over the network;

forward the request from the customer to a home services coordinator over the network;

use the reputation system component to assist the home services coordinator to select a service provider based on customer needs, customer preferences, and a reputation of the service provider;

provide information to the selected service provider over the network sufficient to permit the service provider to select a home delivery provider that can satisfy customer needs;

use the access control component to provide the home delivery provider with access to customer data and with access to a customer physical system over the network; and

request feedback from the customer after the service has been delivered, and use the customer feedback in the reputation system component to update the customer preferences and the reputation of the service provider.

18. The home service coordination platform of claim **17**, further comprising an identity management component.

19. The home services coordination platform of claim **18**, wherein the one or more servers are configured to use the access control component and the identity management component to provide physical access to the home of the customer.

20. The home service coordination platform of claim **17**, wherein the one or more servers are configured to use the access control component to provide physical access to the home of the customer.

21. The home service coordination platform of claim **20**, wherein the one or more servers are configured to provide physical access to the home of the customer, to open an electronic lock on a door to the home of the customer.

* * * * *

Bibliography

[1] ABC4Trust. Attribute-based Credentials for Trust. European Union funded project of the 7th Framework Programme. [Online]. Available: https://abc4trust.eu/

[2] James M. Adler, Wei Dai, Richard L. Green and C. Andrew Neff. Computational Details of the VoteHere Homomorphic Election System. In Proceedings Ann. Intl Conf. Theory and Application of Cryptology and Information Security, *ASIACRYPT*, 2000.

[3] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, vol 17 n. 6, pages 734–749, 2005.

[4] Isaac Agudo, Carmen Fernández-Gago and Javier López. A multidimensional reputation scheme for identity federations. In *Public Key Infrastructures, Services and Applications*, pages 225–238. Springer, 2010.

[5] Luis Ahn, Manuel Blum, NicholasJ. Hopper and John Langford. Captcha: Using hard ai problems for security. In *Advances in Cryptology - EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 294–311. Springer, 2003.

[6] Waleed A. Alrodhan and Chris J. Mitchell. Addressing privacy issues in CardSpace. In Proceedings of the 3rd International Symposium on Information Assurance and Security (IAS '07), pages 285–291, Manchester, UK, 2007.

[7] Anne Anderson. XACML profile for Role Based Access Control (RBAC). *OASIS Access Control TC committee draft*, 1:13, 2004.

[8] Patricia Arias Cabarcos, Florina Almenárez Mendoza, Andrés Marín López and Daniel Díaz Sánchez. Enabling SAML for Dynamic Identity Federation Management. In *Wireless and Mobile Networking Conference*, IFIP Advances in Information and Communication Technology, pages 173–184. Springer, 2009.

[9] Roberto Aringhieri, Ernesto Damiani, Sabine De Capitani Di Vimercati, Stefano Paraboschi and Pierangelo Samarati. Fuzzy techniques for trust and reputation management in anonymous peer-to-peer systems. Journal of the American Society for Information Science and Technology, vol. 57 n. 4, pages 528–537, 2006.

[10] Mikaël Ates, Francesco Buccafurri, Jacques Fayolle, Gianluca Lax. A warning on how to implement anonymous credential protocols into the information card framework. International Journal of Information Security. Vol 11, n. 1, pages 33–40, 2012.

[11] Luigi Atzori, Antonio Iera and Giacomo Morabito. The internet of things: A survey. *Computer Networks*, vol. 54 n. 15, pages 2787–2805, 2010.

[12] Fenye Bao, Ing-Ray Chen and Jia Guo. Scalable, adaptive and survivable trust management for community of interest based Internet of Things systems. In *IEEE Eleventh International Symposium on Autonomous Decentralized Systems (ISADS)*, pages 1–7, March 2013.

[13] Anirban Basu, Hiroaki Kikuchi and Jaideep Vaidya. Efficient privacy-preserving collaborative filtering based on the weighted slope one predictor. *Journal of Internet Services and Information Security*, vol. 1 n. 4, pages 1–20, 2011.

[14] Vittorio Bertocci, Garrett Serack and Caleb Baker. Understanding Windows CardSpace: An Introduction to the Concepts and Challenges of Digital Identities. Addison-Wesley, Reading, Massachusetts, USA, 2008.

[15] Abhilasha Bhargav-Spantzel, Anna C. Squicciarini and Elisa Bertino. Trust negotiation in identity management. *IEEE Security and Privacy*, vol. 5, pages 55–63, 2007.

[16] Dionysus Blazakis. The Apple Sandbox, 2011.

[17] Zvika Brakerski, Craig Gentry and Vinod Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pages 309–325. ACM, 2012.

[18] Jesús Bobadilla, Fernando Ortega, Antonio Hernando and Abraham Gutiérrez. Recommender systems survey. *Knowledge-Based Systems*, vol. 46, pages 109–132, 2013.

[19] Dan Bogdanov, Margus Niitsoo, Tomas Toft and Jan Willemson. High-performance secure multi-party computation for data mining applications. International Journal of Information Security, vol. 11, n.6, pages 403–418, 2012.

[20] Anton Borg, Martin Boldt and Bengt Carlsson. Simulating malicious users in a software reputation system. In *Secure and Trust Computing, Data Management and Applications*, volume 186 of *Communications in Computer and Information Science*, pages 147–156. Springer, 2011.

[21] Azzedine Boukerche, Li Xu and Khalil El-Khatib. Trust-based security for wireless ad hoc and sensor networks. *Computer Communications*, vol. 30 n. 11-12, pages 2413–2427, 2007.

[22] Stefan A. Brands. Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy. MIT Press, Cambridge, MA, USA, 2000.

[23] Stefan A. Brands, Liesje Demuynck and Bart De Decker. A practical system for globally revoking the unlinkable pseudonyms of unknown users. In Proceedings of the 12th Australasian Conference on Information Security and Privacy, ACISP'07. Springer-Verlag, 2007

[24] Franco Callegati, Walter Cerroni and Marco Ramilli. Man-in-the-Middle Attack to the HTTPS Protocol. IEEE Security & Privacy, vol.7, n.1, pages 78–81, 2009.

[25] Ran Canetti, Uri Feige, Oded Goldreich and Moni Naor. Adaptively secure multi-party computation. In *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing*, STOC '96, pages 639–648, New York, NY, USA. ACM, 1996.

[26] Jan Camenisch and Anna Lysyanskaya. Efficient Non-Transferable Anonymous Multi-Show Credential System with Optional Anonymity Revocation. In Theory and Application of Cryptographic Techniques, EUROCRYPT, 2001.

[27] Jan Camenisch and Els Van Herreweghen. Design and Implementation of the Idemix Anonymous Credential System. In Proceedings of the 9th ACM conference on Computer and Communications Security, 2002.

[28] Jan Camenisch, Ioannis Krontiris, Anja Lehmann, Gregory Neven, Christian Paquin, Kai Rannenberg and Harald Zwingelberg. D2.1 Architecture for Attribute-based Credential Technologies. Deliverable of ABC4Trust European Project, 2011.

[29] David W. Chadwick. *Foundations of Security Analysis and Design*, chapter Federated Identity Management, pages 96–120. Number 5705 in LNCS. Springer, 2009.

[30] David W. Chadwick. and George Inman. Attribute Aggregation in Federated Identity Management. IEEE Computer Society, vol.42, n. 5, pages 33–40, 2009.

[31] David Chappell. Introducing Windows CardSpace. MSDN. 2006. Available: http://msdn.microsoft.com/en-us/library/aa480189.aspx

[32] Dong Chen, Guiran Chang, Dawei Sun, Jiajia Li, Jie Jia and Xingwei Wang. TRM-IoT: A trust management model based on fuzzy reputation for internet of things. *Computer Science and Information Systems*, vol. 8 n. 4, pages 1207–1228, 2011.

[33] Ruichuan Chen, Xuan Chao, Liyong Tang, Jianbin Hu and Zhong Chen. CuboidTrust: A Global Reputation-Based Trust Model in Peer-to-Peer Networks. In *Autonomic and Trusted Computing*, number 4610 in LNCS, pages 203–215, Hong Kong, China, jul 2007. 4th International Conference, Springer, ATC 2007.

[34] Pau-Chen Cheng, Pankaj Rohatgi, Claudia Keser, Paul A. Karger, Grant M. Wagner, and Angela Schuett Reninger. Fuzzy multi-level security: An experiment on quantified risk-adaptive access control. In *SP '07: Proceedings of the 2007 IEEE Symposium on Security and Privacy*, pages 222–230, Washington, DC, USA. IEEE Computer Society, 2007.

[35] Daeseon Choi, Seung-Hun Jin and Hyunsoo Yoon. Trust management for user-centric identity management on the internet. In *IEEE International Symposium on Consumer Electronics*, pages 1–4, 2007.

[36] Richard Chow, Manas A. Pathak and Cong Wang. A practical system for privacy-preserving collaborative filtering. In *12th IEEE International Conference on Data Mining Workshops, ICDM Workshops*, pages 547–554. IEEE Computer Society, 2012.

[37] Mihai Christodorescu, Reiner Sailer, Douglas Lee Schales, Daniele Sgandurra and Diego Zamboni. Cloud security is not (just) virtualization security: a short paper. In Proceedings of the 2009 ACM workshop on Cloud computing security (CCSW '09). ACM, New York, NY, USA, pages 97–102, 2009.

[38] Jean-Sébastien Coron, David Naccache and Mehdi Tibouchi. Public key compression and modulus switching for fully homomorphic encryption over the integers. In *Advances in Cryptology–EUROCRYPT 2012*, pages 446–464. Springer, 2012.

[39] Jean-Sebastien Coron and Mehdi Tibouchi. Implementation of the DGHV fully homomorphic encryption scheme. https://github.com/coron/fhe, 2012.

[40] Ivan Damgård, Valerio Pastro, Nigel Smart and Sarah Zakarias. Multiparty computation from somewhat homomorphic encryption. In *Advances in Cryptology–CRYPTO 2012*, pages 643–662. Springer, 2012.

[41] Luca De Alfaro, Ashutosh Kulshreshtha, Ian Pye and B. Thomas Adler. Reputation systems for open collaboration. *Commun. ACM*, vol. 54 n. 8, pages 81–87, 2011.

[42] Jan De Clercq. Single sign-on architectures. In InfraSec '02: Proceedings of the International Conference on Infrastructure Security, pages 40–58, Bristol, UK. Springer-Verlag, 2002.

[43] José Valente de Oliveira. Semantic constraints for membership function optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, vol. 29 n. 1, pages 128–138, 1999.

[44] Marnix Dekker and Giles Hogben. Appstore Security: 5 lines of defence against malware. European Network and Information Security Agency (ENISA). September 2011.

[45] Mina Deng and Kim Wuyts and Riccardo Scandariato and Bart Preneel and Wouter Joosen. A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements. Requirements Engineering, vol. 16 n. 1, pages 3–32, 2010.

[46] Tassos Dimitriou and Antonis Michalas. Multi-party trust computation in decentralized environments. In *5th International Conference on New Technologies, Mobility and Security (NTMS)*, pages 1–5. IEEE, 2012.

[47] Ginés Dólera Tormo and Félix Gómez Mármol. System and method for determining a reputation mechanism, 08 Patent WO2013117224 (A1), 15/08/2013.

[48] Ginés Dólera Tormo, Félix Gómez Mármol, Joao Girao and Gregorio Martínez Pérez. Identity Management: In privacy we trust. Bridging the trust gap in e-Health environments. *IEEE Security & Privacy*, 2013.

[49] Ginés Dólera Tormo, Félix Gómez Mármol and Gregorio Martínez Pérez. Dynamic and flexible selection of a reputation mechanism for heterogeneous environments. *Future Generation Computer Systems*, June 2014.

[50] Ginés Dólera Tormo, Félix Gómez Mármol and Gregorio Martínez Pérez. On the Application of Trust and Reputation Management and User-centric Techniques for Identity Management Systems. XII Spanish Meeting on Cryptology and Information Security (RECSI 2012), San Sebastián, Spain (2012).

[51] Ginés Dólera Tormo, Félix Gómez Mármol and Gregorio Martínez Pérez. ROMEO: ReputatiOn Model Enhancing OpenID Simulator. In 19th European Symposium on Research in Computer Security (ESORICS), Security & Trust Management Workshop (STM 2014), Wroclaw, Poland, 7-11/09/2014. LNCS 8743, pages 193–197, September 2014.

[52] Ginés Dólera Tormo, Félix Gómez Mármol and Gregorio Martínez Pérez. Towards the Integration of Reputation Management in OpenID. *Special Issue on Secure Mobility in Future Communication Systems under Standardization, Computer Standards & Interfaces*, vol. 36 n. 3, pages 438–453, March 2014.

[53] Ginés Dólera Tormo, Félix Gómez Mármol and Gregorio Martínez Pérez. Identity Management in Cloud Systems. *Security, Privacy and Trust in Cloud Systems*, Eds: S. Nepal, M. Pathan, Publisher: Springer, ISBN: 978-3-642-38585-8, pp 177-210, 2014

[54] Ginés Dólera Tormo, Félix Gómez Mármol and Gregorio Martínez Pérez. Towards privacy-preserving reputation management for hybrid broadcast broadband applications. *Computers & Security*, 2014.

[55] Ginés Dólera Tormo, Gabriel López Millán and Gregorio Martínez Pérez. Definition of an advanced identity management infrastructure. *International Journal of Information Security*, vol. 12, n. 3, pages 173–200, 2013.

[56] John R. Douceur and Judith S. Donath. The sybil attack. In *Proceedings for the 1st International Workshop on Peer-to-Peer Systems (IPTPS '02)*, pages 251–260, 2002.

[57] Wenliang Du and Zhijun Zhan. A practical approach to solve secure multi-party computation problems. In *Proceedings of the 2002 Workshop on New Security Paradigms*, NSPW '02, pages 127–135, New York, NY, USA. ACM, 2002.

[58] Eclipse Foundation. Higgins 2.0 Personal Data Service. [Online]. Available: http://www.eclipse.org/higgins/, 2007

[59] Taher ElGamal. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. In *Proceedings of CRYPTO 84 on Advances in cryptology*, number 4, pages 469–472, jul 1985.

[60] Marlena Erdos and Scott Cantor. Shibboleth-Architecture DRAFT v05. Internet2/MACE, Available: http://shibboleth.internet2.edu/docs/draft-internet2-shibboleth-arch-v05.pdf, 2002.

[61] Zekeriya Erkin, Michael Beye, Thijs Veugen and Reginald L. Lagendijk. Privacy-preserving content-based recommender system. In *Proceedings of the on Multimedia and Security*, MM&Sec '12, pages 77–84. ACM, 2012.

[62] Mohd Farhan Md Fudzee and Jemal Abawajy. *Network and Traffic Engineering in Emerging Distributed Computing Applications*, chapter Management of Service Level Agreement for Service-Oriented Content Adaptation Platform, pages 21–42. Information Science Reference, 2012.

[63] Randy Farmer and Bryce Glass. *Building Web Reputation Systems*. O'Reilly Media, March 2010.

[64] Adrienne Porter Felt, Matthew Finifter, Erika Chin, Steve Hanna and David Wagner. A survey of mobile malware in the wild. In *Proceedings of the 1st ACM Workshop on Security and Privacy in Smartphones and Mobile Devices*, SPSM '11, pages 3–14. ACM, 2011.

[65] Carol Fung, Jie Zhang, Issam Aib and Raouf Boutaba. Trust management and admission control for host-based collaborative intrusion detection. *Journal of Network and Systems Management*, vol. 19 n. 2, pages 257–277, 2010.

[66] Sebastian Gajek, Jörg Schwenk, Michael Steiner and Chen Xuan. Risks of the CardSpace Protocol. Lecture Notes in Computer Science, vol. 5735 pages 278–293, 2009.

[67] Nurit Gal-Oz, Niv Gilboa and Ehud Gudes. Schemes for privately computing trust and reputation. In *Trust Management IV - 4th IFIP WG 11.11 International Conference, IFIPTM 2010*, pages 1–16, 2010.

[68] Craig Gentry. *A fully homomorphic encryption scheme.* PhD thesis, Stanford University, 2009.

[69] Craig Gentry and Shai Halevi. Implementing gentry's fully-homomorphic encryption scheme. In *Proceedings of the 30th Annual international conference on Theory and applications of cryptographic techniques: advances in cryptology*, EUROCRYPT'11, pages 129–148, Berlin, Heidelberg, 2011. Springer-Verlag.

[70] Manuel Gil Pérez, Félix Gómez Mármol, Gregorio Martínez Pérez and Antonio F. Skarmeta Gómez. RepCIDN: A Reputation-based Collaborative Intrusion Detection Network to Lessen the Impact of Malicious Alarms. *Journal of Network and Systems Management*, vol 21 n. 1, pages 128–167, 2013.

[71] Joao Girao, Brigitta Lange, Nils Gruschka, Ginés Dólera Tormo and Félix Gómez Mármol. Method to support an advanced home services coordination platform, US 20130304488 A1, 14/11/2013.

[72] Oded Goldreich, Silvio Micali and Avi Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. Journal of the ACM (JACM) 38, n. 3, pages 690–728, 1991.

[73] Shafi Goldwasser, Silvio Micali and Charles Rackoff. The knowledge complexity of interactive proof systems. SIAM Journal on computing 18, n. 1, pages 186–208, 1989.

[74] Félix Gómez Mármol, Marcus Quintino Kuhnen and Gregorio Martínez Pérez. Enhancing OpenID through a Reputation Framework. In *Proceedings of the 8th international conference on Autonomic and Trusted Computing, ATC'11*, number 6906 in LNCS, pages 1–18. Springer-Verlag, 2011.

[75] Félix Gómez Mármol, Joao Girao and Gregorio Martínez Pérez. TRIMS, a Privacy-aware Trust and Reputation Model for Identity Management Systems. *Elsevier Computer Networks Journal*, vol 54 n. 16, pages 2899–2912, 2010.

[76] Félix Gómez Mármol, Javier Gómez Marín-Blázquez and Gregorio Martínez Pérez. Linguistic Fuzzy Logic Enhancement of a Trust Mechanism for Distributed Networks. In *Proceedings of the Third IEEE International Symposium on Trust, Security and Privacy for Emerging Applications (TSP-10)*, pages 838–845, Bradford, UK, 2010.

[77] Félix Gómez Mármol and Gregorio Martínez Pérez. Towards Pre-Standardization of Trust and Reputation Models for Distributed and Heterogeneous Systems. *Computer Standards & Interfaces, Special Issue on Information and Communications Security, Privacy and Trust: Standards and Regulations*, vol. 32 n. 4, pages 185–196, 2010.

[78] Félix Gómez Mármol and Gregorio Martínez Pérez. Providing Trust in Wireless Sensor Networks using a Bio-Inspired Technique. *Telecommunication Systems Journal*, vol. 46 n. 2, pages 163–180, 2011.

[79] Félix Gómez Mármol and Gregorio Martínez Pérez. Security Threats Scenarios in Trust and Reputation Models for Distributed Systems. *Elsevier Computers & Security*, vol. 28 n. 7, pages 545–556, October 2009.

[80] Félix Gómez Mármol and Gregorio Martínez Pérez. State of the art in trust and reputation models in P2P networks. In Xuemin Shen, Heather Yu, John Buford and Mursalin Akon, editors, *Handbook of Peer-to-Peer Networking*, pages 761–784. Springer US, 2010.

[81] Félix Gómez Mármol, Gregorio Martínez Pérez and Antonio F. Gómez Skarmeta. TACS, a Trust Model for P2P Networks. *Wireless Personal Communications, Special Issue on "Information Security and data protection in Future Generation Communication and Networking"*, vol. 51 n. 1, pages 153–164, 2009.

[82] Félix Gómez Mármol, Gregor Rozinaj, Sebastian Schumann, Ondrej Lábaj, and Juraj Kacur. Smart AppStore: expanding the frontiers of Smartphone ecosystems. *IEEE Computer*, vol. 47, n. 6, pages 42–47, 2014.

[83] Paolo Guarda, Nicola Zannone. Towards the development of privacy-aware systems. Information and Software Technology, vol. 51 n. 2, pages 337–350, 2009.

[84] Ehud Gudes, Nurit Gal-Oz and Alon Grubshtein. Methods for computing trust and reputation while preserving privacy. *Data and Applications Security XXIII*, pages 291–298, 2009.

[85] Ido Guy, Naama Zwerdling, David Carmel, Inbal Ronen, Erel Uziel, Sivan Yogev, and Shila Ofek-Koifman. Personalized recommendation of social software items based on social relations. In *Proceedings of the third ACM conference on Recommender systems*, pages 53–60. ACM, 2009.

[86] Eran Hammer-Lahav and David Recordon. The OAuth 1.0 Protocol. Internet Engineering Task Force (IETF) RFC 5849, 2010.

[87] Edward Hamilton, Mischa Kriens, Helen Karapandyic, Karim Yaici and Mark Main. Report on trust and reputation models, December 2011.

[88] Jiawei Han and Kamber Micheline. *Data Mining: Concepts and Techniques.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.

[89] Chung-Wei Hang and Munindar P. Singh. Selecting trustworthy service in service-oriented environments. In *The 12th AAMAS Workshop on Trust in Agent Societies*, May 2009.

[90] Marit Hansen, Ari Schwartz and Alissa Cooper. Privacy and identity management. *Security & Privacy, IEEE*, vol. 6 n. 2, pages 38–45, 2008.

[91] Patrick Harding, Paul Madsen, Trae Drake and Chuck Mortimore. System for Cross-Domain Identity Management: Core Schema. Internet Draft. draft-ietf-scim-core-schema-00 (SCIM), 2012.

[92] Dick Hardt. The OAuth 2.0 Authorization Framework. Technical report, IETF. Available: http://tools.ietf.org/html/draft-ietf-oauth-v2-31, 2012.

[93] Javier Herranz, Javier Iñigo and Helena Pujol. Privacy Features of Authentication Systems. Proceeding of the First Workshop on Law and Web 2.0, Barcelona, Spain. pages. 35–46, 2009.

[94] Adrian Holzer and Jan Ondrus. Trends in mobile application development. In *MOBILWARE Workshops*, pages 55–64, 2009.

[95] Adrian Holzer and Jan Ondrus. Mobile application market: A developer's perspective. *Telematics and Informatics*, vol. 28 n. 1, pages 22–31, 2011.

[96] Wolfgang Hoschek, Javier Jaen-Martinez, Asad Samar, Heinz Stockinger and Kurt Stockinger. Data Management in an International Data Grid Project. Lecture Notes in Computer Science. Vol. 1971, pages 77–90, 2000.

[97] Nathaniel Husted, Hassen Saïdi and Ashish Gehani. Smartphone security limitations: Conflicting traditions. In *Proceedings of the 2011 Workshop on Governance of Technology, Information, and Policies*, GTIP '11, pages 5–12. ACM, 2011.

[98] Chunhua Hu and Jibo Liu. Web services composition approach based on trust computing mode. In *5th IEEE Asia-Pacific Services Computing Conference, APSCC 2010*, pages 663–670, December 2010.

[99] Chenlin Huang, Huaping Hu and Zhiying Wang. A dynamic trust model based on feedback control mechanism for P2P applications. In *Autonomic and Trusted Computing*, number 4158 in LNCS, pages 312–321, Wuhan, China, sep 2006. Springer.

[100] Hybrid Broadcast Broadband TV (HbbTV). http://www.hbbtv.org/, 2013.

[101] IBM Research, Zurich. Specification of the Identity Mixer Cryptographic Library, 2010.

[102] Identity Commons. [Online]. Available: http://www.identitycommons.net/.

[103] J. S. Roger Jang and C. T. Sun. Functional equivalence between radial basis function networks and fuzzy inference systems. *IEEE Transactions on Neural Networks*, vol 4 n. 1, pages 156–159, January 1993.

[104] Tom N. Jagatic, Nathaniel A. Johnson, Markus Jakobsson and Filippo Menczer. Social phishing. Communications of the ACM 50, pages 94–100, 2007.

[105] Slinger Jansen, Anthony Finkelstein and Sjaak Brinkkemper. A sense of community: A research agenda for software ecosystems. In *ICSE Companion*, pages 187–190. IEEE, 2009.

[106] Jostein Jensen. Benefits of federated identity management: a survey from an integrated operations viewpoint. In *Proceedings of the IFIP WG 8.4/8.9 international cross domain conference on availability, reliability and security for business, enterprise and health information systems*, ARES'11, pages 1–12. Springer, 2011.

[107] Michael B. Jones and Michael McIntosh. Identity Metasystem Interoperability Version 1.0 (IMI 1.0). OASIS Standard, 2009.

[108] Michael B. Jones. The identity metasystem: A user-centric, inclusive web authentication solution. Toward a More Secure Web-W3C Workshop on Transparency and Usability of Web Authentication, 2006.

[109] Audun Jøsang, John Fabre, Brian Hay, James Dalziel and Simon Pope. Trust requirements in identity management. In *Proceedings of the 2005 Australasian workshop on Grid computing and e-research-Volume 44*, pages 99–108. Australian Computer Society, Inc., 2005.

[110] Audun Jøsang, Roslan Ismail and Colin Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, vol. 43 n. 2, pages 618–644, 2007.

[111] Seny Kamara, Payman Mohassel and Mariana Raykova. Outsourcing multi-party computation. Cryptology ePrint Archive, Report 2011/272, 2011. http://eprint.iacr.org/.

[112] Sepandar D. Kamvar, Mario T. Schlosser and Hector García-Molina. The eigentrust algorithm for reputation management in p2p networks. In *Proceedings of the 12th International Conference on World Wide Web*, WWW '03, pages 640–651, New York, NY, USA, 2003. ACM.

[113] Kantara Initiative. [Online]. Available: http://kantarainitiative.org/.

[114] Alexandros Karatzoglou, Linas Baltrunas, Karen Church and Matthias Böhmer. Climbing the app wall: Enabling mobile app discovery through context-aware recommendations. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 2527–2530. ACM, 2012.

[115] Reid Kerr and Robin Cohen. Smart cheaters do prosper: defeating trust and reputation systems. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 993–1000. International Foundation for Autonomous Agents and Multiagent Systems, 2009.

[116] M. Kolšek. Session fixation vulnerability in web-based applications. ACROS Security. Available: http://www.acrossecurity.com/papers/session_fixation.pdf, 2002.

[117] Srividya Kona, Ajay Bansal and M. Brian Blake. Trust-based dynamic web service composition using social network analysis. In *IEEE International Workshop on Business Applications for Social Network Analysis (BASNA 2010)*, December 2010.

[118] Georgios Kontaxis, Michalis Polychronakis and Evangelos P. Markatos. Minimizing information disclosure to third parties in social login platforms. International Journal of Information Security. Vol 11, n. 5, pages 321–332, 2012.

[119] Hyea Kyeong Kim, Jae Kyeong Kim and Young U Ryu. Personalized recommendation over a customer network for ubiquitous shopping. *IEEE Transactions on Services Computing*, vol. 2 n. 2, pages 140–151, 2009.

[120] Steven L Kinney. *Trusted platform module basics: using TPM in embedded systems.* Newnes, 2006.

[121] Gerd Kortuem, Fahim Kawsar, Daniel Fitton and Vasughi Sundramoorthy. Smart objects as building blocks for the internet of things. *Internet Computing, IEEE*, vol. 14 n. 1, pages 44–51, 2010.

[122] Andreas Leicher, Andreas U Schmidt, Yogendra Shah and Inhyok Cha. Trusted computing enhanced user authentication with openid and trustworthy user interface. *International Journal of Internet Technology and Secured Transactions*, vol. 3 n. 4, pages 331–353, 2011.

[123] Cane Wing-Ki Leung, Stephen Chi-Fai Chan, Fu-Lai Chung and Grace Ngai. A probabilistic rating inference framework for mining user preferences from reviews. *World Wide Web*, vol. 14 n. 2, pages 187–215, 2011.

[124] Jingtao Li, Yinan Jing, Xiaochun Xiao, Xueping Wang and Gendu Zhang. A trust model based on similarity-weighted recommendation for P2P environments. *Journal of Software*, vol 18 n. 1, pages 157–167, 2007.

[125] Kwei-Jay Lin, Haiyin Lu, Tao Yu and Chia-en Tai. A reputation and trust management broker framework for web applications. In *In International Conference on e-Technology, e-Commerce, and e-Services*, pages 262–269. IEEE Computer Society, 2005.

[126] Nai-Wei Lo and Hsiao-Chien Tsai. A reputation system for traffic safety event on vehicular ad hoc networks. *EURASIP Journal on Wireless Communications and Networking*, vol. 2009, pages 1–9, April 2009.

[127] Adriana López-Alt, Eran Tromer and Vinod Vaikuntanathan. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In *Proceedings of the 44th symposium on Theory of Computing*, pages 1219–1234. ACM, 2012.

[128] Eve Maler and Drummond Reed. The venn of identity: Options and issues in federated identity management. IEEE Security & Privacy, 6, pages 16–23, 2008.

[129] Zaki Malik and Athman Bouguettaya. Reputation bootstrapping for trust establishment among web services. *IEEE Internet Computing*, vol. 13, pages 40–47, 2009.

[130] Steve Mansfield-Devine. Paranoid android: just how insecure is the most popular mobile platform? *Network Security*, vol. 2012 n. 9, pages 5–10, 2012.

[131] Sergio Marti and Hector García-Molina. Identity crisis: anonymity vs reputation in P2P systems. In *Proceedings for the Third International Conference on Peer-to-Peer Computing (P2P 2003)*, pages 134–141, Linköping, Sweden, sep 2003.

[132] CarloMaria Medaglia and Alexandru Serbanati. An overview of privacy and security issues in the internet of things. In Daniel Giusto, Antonio Iera, Giacomo Morabito and Luigi Atzori, editors, *The Internet of Things*, pages 389–395. Springer New York, 2010.

[133] Teresa Mendyk-Krajewska, Zygmunt Mazur and Hanna Mazur. Threats to wireless technologies and mobile devices and company network safety. In Springer, editor, *Internet - Technical Developments and Applications 2*, volume 118 of *Advances in Intelligent and Soft Computing*, pages 209–225, 2012.

[134] Apurva Mohan and Douglas M. Blough. AttributeTrust - a framework for evaluating trust in aggregated attributes via a reputation system. In *Proceedings of the 2008 Sixth Annual Conference on Privacy, Security and Trust*, pages 201–212, 2008.

[135] Mohammad Momani and Subhash Challa. Survey of trust models in different network domains. International Journal of Ad hoc, Sensor & Ubiquitous Computing, vol. 1 n. 3, pages 1–19, 2010.

[136] Michele Mostarda, Davide Palmisano, Federico Zani and Simone Tripodi. Towards an openid-based solution to the social network interoperability problem. In *W3C workshop on the future of social networking*, pages 15–16, 2009.

[137] Michael Naehrig, Kristin Lauter and Vinod Vaikuntanathan. Can homomorphic encryption be practical? In *Proceedings of the 3rd ACM Workshop on Cloud Computing Security Workshop*, CCSW '11, pages 113–124, New York, NY, USA, 2011. ACM.

[138] Arun Nanda and Michael B. Jones. Identity Selector Interoperability Profile v1. 5. Microsoft Corporation, Available: http://download.microsoft.com/download/1/1/a/11ac6505-e4c0-4e05-987c-6f1d31855cd2/Identity_Selector_Interoperability_Profile_V1.5.pdf, 2008.

[139] B Clifford Neuman and Theodore Ts'o. Kerberos: An authentication service for computer networks. *Communications Magazine, IEEE*, vol. 32 n. 9, pages 33–38, 1994.

[140] Rishab Nithyanand and Karthik Raman. Fuzzy privacy preserving peer-to-peer reputation management. Technical report, Cryptology ePrint Archive, Report 2009/442, 2009.

[141] OASIS IDCloud TC. OASIS Identity in the Cloud TC. [Online]. Available: http://wiki.oasis-open.org/id-cloud/

[142] OASIS Privacy Management Reference Model (PMRM) TC [Online]. Available: http://www.oasis-open.org/committees/pmrm

[143] OASIS Standard. Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) version 2.0, 2005.

[144] OASIS Standard. eXtensible Access Control Markup Language TC v2.0 (XACML). Available: http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf, 2005

[145] OASIS Standard. Open reputation management systems (ORMS). http://www.oasis-open.org/committees/orms, 2008.

[146] OAuth Community. [Online]. Available: http://oauth.net/community/

[147] Hyun-Kyung Oh and Seung-Hun Jin. The Security Limitations of SSO in OpenID. In *10th International Conference on Advanced Communication Technology*, volume 3, pages 1608–1611, Feb 2008.

[148] Mawloud Omar, Yacine Challal and Abdelmadjid Bouabdallah. Reliable and fully distributed trust model for mobile ad hoc networks. *Computers and Security*, vol. 28 n. 3-4, pages 199–214, 2009.

[149] OpenID Community. [Online]. Available: http://openid.net/community/

[150] Christian Paquin and Greg Thompson. U-Prove CTP White Paper. Microsoft Technical Report, 2010.

[151] Sharon Paradesi, Prashant Doshi and Sonu Swaika. Integrating behavioral trust in web service compositions. In *Proceedings of the 2009 IEEE International Conference on Web Services, ICWS '09*, pages 453–460, 2009.

[152] Pankesh Patel, Ajith Ranabahu and Amit Sheth. Service level agreement in cloud computing. In *Cloud Workshops at OOPSLA*, 2009.

[153] Anand Patwardhan, Anupam Joshi, Tim Finin and Yelena Yesha. A Data Intensive Reputation Management Scheme for Vehicular Ad Hoc Networks. In *Proceedings of the Second International Workshop on Vehicle-to-Vehicle Communications*, pages 1–8, July 2006.

[154] Pramod S. Pawar, Muttukrishnan Rajarajan, S Krishnan Nair and Andrea Zisman. Trust model for optimized cloud services. In *Trust Management VI*, pages 97–112. Springer, 2012.

[155] Siani Pearson and Azzedine Benameur. Privacy, Security and Trust Issues Arising from Cloud Computing. Proceedings of the Second International Conference on Cloud Computing Technology and Science (CloudCom). Bristol, UK. pages 693–702, 2010.

[156] Witold Pedrycz and Fernando Gomide. *An introduction to Fuzzy Sets: Analysis and Design*. The MIT Press, Cambridge, Masssachusetts, USA, 1998.

[157] Fernando Pereñíguez GarcÃa, Rafael Marín-López, Georgios Kambourakis, Antonio Ruiz-Martínez, Stefanos Gritzalis and Antonio F. Skarmeta-Gómez. KAMU: providing advanced user privacy in Kerberos multi-domain scenarios. *International Journal of Information Security*, pages 1–21, 2013.

[158] Heloise Pieterse and Martin S. Olivier. Android botnets on the rise: Trends and characteristics. In *Information Security for South Africa*, pages 1–5. IEEE, 2012.

[159] PrimeLife. European Union funded project of the 7th Framework Programme. [Online]. Available: http://primelife.ercim.eu/.

[160] Steve Quirolgico and Jeffrey Voas and Rick Kuhn. Vetting Mobile Apps. *IT Professional*, vol. 13, pages 9–11, July/August 2011.

[161] David Recordon and Drummond Reed. OpenID 2.0: a platform for user-centric identity management. In *Proceedings of the second ACM workshop on Digital identity management, DIM '06*, pages 11–16, 2006.

[162] Paul Resnick and Richard Zeckhauser. Trust among strangers in internet transactions: Empirical analysis of ebay's reputation system. *The economics of the Internet and E-commerce*, 11:127, 2002.

[163] Paul Resnick, Richard Zeckhauser, John Swanson and Kate Lockwood. The value of reputation on eBay: A controlled experiment. *Experimental Economics*, vol. 9 n. 2, pages 79–101, 2006.

[164] Sebastian Ries, Marc Fischlin, Leonardo A Martucci and M Muhlhauser. Learning whom to trust in a privacy-friendly way. In *IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 214–225. IEEE, 2011.

[165] Rodrigo Roman, Pablo Najera and Javier Lopez. Securing the internet of things. *Computer*, vol. 44 n. 9, pages 51–58, 2011.

[166] Nat Sakimura. Coping with information asymmetry. In *Identity Management Conference, SESSION G: Managing Risk & Reducing Online Fraud Using New Security Technologies*, pages 1–14, Washington, US, Sep 2010. OASIS.

[167] Anil Saldhana, Anthony Nadalin and Matt Rutkowski. Identity in the Cloud Use Cases Version 1.0. Available: http://docs.oasis-open.org/id-cloud/IDCloud-usecases/v1.0/cn01/IDCloud-usecases-v1.0-cn01.html, May 2012.

[168] Simon SY Shim, Geetanjali Bhalla and Vishnu Pendyala. Federated identity management. *Computer*, vol 38 n. 12, pages 120–122, 2005.

[169] Jansen Slinger and Bloemendal Ewoud. Defining app stores: The role of curated marketplaces in software ecosystems. In *ICSOB*, pages 195–206, 2013.

[170] Don Smith. The challenge of federated identity management. *Network Security*, 2008.

[171] Sandra Steinbrecher. Enhancing Multilateral Security in and by Reputation Systems. In *proceedings of the IFIP/FIDIS Internet Security and Privacy Summer School, Brno 2008*, volume 298, pages 135–150. Springer Berlin Heidelberg, September 2008.

[172] STORK (Secure idenTity acrOss boRders linKed). European Union funded project of the 7th Framework Programme. [Online]. Available: https://www.eid-stork.eu/.

[173] SWIFT (Secure Widespread Identities for Federated Telecommunications). European Union funded project of the 7th Framework Programme. [Online]. Available: http://www.ist-swift.org/.

[174] Sysmesh Ltd. Truster. http://www.truster.org/, 2008.

[175] Hassan Takabi, James B. D. Joshi and Gail-Joon Ahn. Security and privacy challenges in cloud computing environments. *IEEE Security and Privacy*, vol. 8, pages 24–31, 2010.

[176] Lu Tan and Neng Wang. Future internet: The internet of things. In *3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)*, vol. 5, pages 376–380, 2010.

[177] A. S. Tanenbaum and M. Van Steen. Distributed systems: principles and paradigms. Prentice Hall, 2001.

[178] Mozhgan Tavakolifard and Kevin C. Almeroth. Social computing: an intersection of recommender systems, trust/reputation systems, and social networks. *IEEE Network*, vol. 26 n. 4, pages 53–58, 2012.

[179] Bart van Delft and Martijn Oostdijk. A security analysis of OpenID. Policies and Research in Identity Management vol. 343, pages 73–84, 2010.

[180] The White House. National Strategy for Trusted Identities in Cyberspace (NSTIC). [Online]. Available: http://www.nist.gov/.

[181] Paul Trevithick. Relationship Cards. Higgins Report. Available: http://www.eclipse.org/higgins/documents/relationship-cards.html, 2009.

[182] U-Prove: Microsoft Corporation Technology. [Online]. Available: http://www.microsoft.com/u-prove, 2010.

[183] Berkant Ustaoğlu. Integrating identity-based and certificate-based authenticated key exchange protocols. International Journal of Information Security. Vol. 10, n.4, pages 201–212, 2011.

[184] Marten Van Dijk, Craig Gentry, Shai Halevi and Vinod Vaikuntanathan. Fully homomorphic encryption over the integers. In *Advances in Cryptology–EUROCRYPT 2010*, pages 24–43. Springer, 2010.

[185] Luis von Ahn, Manuel Blum and John Langford. Telling humans and computers apart automatically. *Commun. ACM*, vol. 47 n. 2, pages 56–60, February 2004.

[186] Cong Wang, Qian Wang, Kui Ren and Wenjing Lou. Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing. Proceedings of the 29th conference on Information communications (INFOCOM'10). IEEE Press, Piscataway, NJ, USA, pages 525–533, 2010.

[187] Shouxin Wang, Li Zhang, Shuai Wang and Xiang Qiu. A cloud-based trust model for evaluating quality of web services. *Journal of Computer Science and Technology*, vol. 25, n. 6, pages 1130–1142, 2010.

[188] Yuan Wang, Ye Tao, Ping Yu, Feng Xu and Jian Lu. A Trust Evolution Model for P2P Networks. In *Autonomic and Trusted Computing*, number 4610 in LNCS, pages 216–225, Hong Kong, China, jul 2007. 4th International Conference, ATC 2007, Springer.

[189] Thomas Wason, Scott Cantor, Jeff Hodges, John Kemp and Peter Thompson. Liberty Id-FF Architecture Overview. Liberty Alliance, 2003.

[190] Web Identity Working group. [Online]. Available: http://www.w3.org/2011/08/webidentity-charter.html.

[191] Phillip J. Windley, Devlin Daley, Bryant Cutler and Kevin Tew. Using reputation to augment explicit authorization. In *Proceedings of the 2007 ACM workshop on Digital identity management, DIM '07*, pages 72–81, 2007.

[192] Phillip J. Windley, Kevin Tew and Devlin Daley. A framework for building reputation systems. In *Proceedings of the Sixteenth International World Wide Web Conference, WWW2007*, pages 1–10, Banff, Canada, may 2007.

[193] Jennifer Yicka, Biswanath Mukherjeea and Dipak Ghosal. A survey on sensor networks. *Computer Networks*, vol. 52 n. 12, pages 2292–2330, Aug 2008.

[194] Josh Jia-Ching Ying, Eric Hsueh-Chan Lu, Wang-Chien Lee, Tz-Chiao Weng and Vincent S Tseng. Mining user similarity from semantic trajectories. In *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks*, pages 19–26. ACM, 2010.

[195] Lofti A. Zadeh. Fuzzy sets. *Information and Control*, vol. 8, pages 338–353, 1965.

[196] Lotfi A. Zadeh. From computing with numbers to computing with words—from manipulation of measurements to manipulation of perceptions. volume 573, pages 36–58. AIP, 2001.

[197] Huanyu Zhao and Xiaolin Li. H-trust: A robust and lightweight group reputation system for peer-to-peer desktop grid. In *28th International Conference on Distributed Computing Systems Workshops ICDCS '08*, pages 235–240, June 2008.

[198] Yajin Zhou, Zhi Wang, Wu Zhou and Xuxian Jiang. Hey, you, get off of my market: Detecting malicious apps in official and alternative android markets. In *Proceedings of the 19th Annual Network and Distributed System Security Symposium*, 2012.

[199] Cai-Nicolas Ziegler and Jennifer Golbeck. Investigating interactions of trust and interest similarity. *Decision Support Systems (Emerging Issues in Collaborative Commerce)*, vol. 43 n. 2, pages 460–475, 2007.