Wireless Pers Commun (2014) 75:1769–1786
DOI 10.1007/s11277-013-1338-y

To Federate or Not To Federate: A Reputation-Based Mechanism to Dynamize Cooperation in Identity Management

Patricia Arias Cabarcos · Florina Almenárez · Félix Gómez Mármol · Andrés Marín

Published online: 1 August 2013

Abstract Identity Management systems cannot be centralized anymore. Nowadays, users have multiple accounts, profiles and personal data distributed throughout the web and hosted by different providers. However, the online world is currently divided into identity silos forcing users to deal with repetitive authentication and registration processes and hindering a faster development of large scale e-business. Federation has been proposed as a technology to bridge different trust domains, allowing user identity information to be shared in order to improve usability. But further research is required to shift from the current static model, where manual bilateral agreements must be pre-configured to enable cooperation between unknown parties, to a more dynamic one, where trust relationships are established on demand in a fully automated fashion. This paper presents IdMRep, the first completely decentralized reputation-based mechanism which makes dynamic federation a reality. Initial experiments demonstrate its accuracy as well as an assumable overhead in scenarios with and without malicious nodes.

Keywords Identity management · Trust and reputation management · Identity federation · Cooperative systems

P. Arias Cabarcos · F. Almenárez · A. Marín
Universidad Carlos III de Madrid, 28911 Leganés, Madrid, Spain
e-mail: ariasp@it.uc3m.es

F. Almenárez
e-mail: florina@it.uc3m.es

A. Marín
e-mail: amarin@it.uc3m.es

F. Gómez Mármol (✉)
NEC Laboratories Europe, 69115 Heidelberg, Germany
e-mail: felix.gomez-marmol@neclab.eu

1 Introduction

1.1 Motivation

Identity Federation has emerged as a key concept for identity management (IdM), as it constitutes the basis to reduce complexity in the companies and improve users' experience. Its main goal is to share and distribute identity information across different domains according to established policies. Thus, the federation model enables roaming users of one domain to securely access resources of another domain seamlessly, without the need for redundant user login processes [1].

Particularly, the most popular use-case in Federated Identity Management (FIM) is Single Sign-On (SSO), which allows mobile wireless users to authenticate at a single domain and gain access to multiple ones without providing additional information. This feature becomes more and more demanded as the number of available applications grows and the mobility of users increases. Furthermore, other interesting examples based on FIM are: user attribute exchange, user account provisioning, wireless users roaming or entitlement management [1].

Due to the importance of the federation paradigm for online IdM [2], a lot of work has been done so far. As a result, the industry and research community have produced a number of standards and specifications [3–8] representing the fundamental building blocks to accomplish identity federation. However, none of the specifications define a suitable trust model to allow the establishment of dynamic federations. More specifically, there are two ways of establishing a federation today according to FIM frameworks:

1. no trust model is involved (i.e., *"accept-all-comers"* philosophy)
2. a rigid trust model is recommended [9]

In the first case, the Service Providers (SPs) will blindly rely on authentication, authorization and attributes of the users sent by every Identity Provider (IdP), and similarly, IdPs will transmit user information to every SP. This solution is flexible but the main problem is security, since we cannot assume that every entity will always behave properly.

The second case, in turn, implies each entity having the pre-configured list of other entities, which are considered trustworthy. Thus, formal agreements are established before any interaction so that an entity could never interact with entities that are not contained in its trust list. Hence, security is achieved but the solution lacks flexibility and scalability [10–12]: a lot of central administration is required and so the initial setup complexity is a high barrier and may not worth adopting these procedures for a short-term collaboration.

Summarizing, trust is a fundamental issue to address scalability, improve security and enhance flexibility, which in turn constitute key elements in open and distributed scenarios, like mobile wireless users (a massive number of roaming users could unexpectedly try to join and login into the same domain). Moreover, the flexibility of every federation framework is tied to the underlying trust model, often poorly defined or even out of the specifications scope [13]. For this reason, new enhanced techniques are required to achieve ad-hoc dynamic federation. Furthermore, the significance of research on this topic has been recently highlighted to the point of stating that "*If dynamic federation negotiation and trust management in IdM systems could be achieved, it would revolutionize the internet marketplace*" [14].

As a solution to the ubiquitous problem of trust in new short-term relationships on wireless communication networks, reputation systems have immediate appeal. Reputation can be combined with other trust data (i.e., history of past interactions), to make smarter decisions [15–17]. However, the application of this dimension of trust to IdM has not been fully addressed yet, and here we aim to evaluate its usefulness to build a reputation-based federated

model. Thought there are related proposals [18,19] and previous work [17], this is the first approach that introduces reputation in FIM without requiring any form of centralized storage.

1.2 Contribution and Outline

With the goal to move from the conventional static bilateral agreements to automated dynamic federation, we propose a reputation-based federation scheme allowing trust relationships to be established on-demand driven by users needs. Accordingly, this paper is organized as follows: Sect. 2 presents the requirements to build a dynamic federation system and proposes a generic architecture for the involved entities. Then, in Sect. 3, we introduce a protocol (IdMRep) for reputation exchange. Furthermore, Sect. 4 shows simulation results obtained after testing the reputation-based model on top of different FIM topologies. Section 5 explains the main related work and, finally, Sect. 6 presents the main conclusions and future research lines. The work presented here constitutes the basis for future enhancements and promising developments in the field of reputation management integration within FIM.

2 Introducing Reputation in FIM Systems

2.1 Architecture for Reputation-Based FIM

We envision FIM as a procedure consisting of two phases [20]:

1. **Pre-Federation Phase**, in which providers establish a relationship, decide on protocols to interoperate, agree on common rules and policies, etc. It can be understood as Bootstrapping, allowing parties to gather information about each other and make decisions about cooperation initiation.
2. **Post-Federation Phase**, which encompasses transactions between two federated entities (e.g., requesting user attributes or accepting authentication claims). Here, entities have basic information to support their decisions: data derived from the Bootstrapping and, if more interactions have occurred, a history of transactions. It can be viewed as the Evolution Phase, since entities progressively construct and consolidate their relationships.

In current FIM systems, Pre-Federation consists of establishing agreements between entities and manually setting up a Circle of Trust (CoT). Then, in Post-Federation, it is assumed that everything will work, since interaction is only possible with the entities that have been pre-configured. The generic architecture diagram of current FIM implementations is shown in Fig. 1, where the CoT configuration component is the module that statically handles trust relationships. The architecture includes also a component to provide support for cryptographic operations, a module that implements identity services (SSO, Single-Log Out or SLO, etc.), and a logging module for registering user and providers activities.

Our goal is to extend and enhance the initial architecture in Fig. 1 in order to allow entities to dynamically move from Pre-Federation to Post-Federation with a certain degree of trust and to continually monitor and consolidate this trust relationship. To this end, the main requirements we identify during these phases are:

- **R1**. Storage, dissemination and aggregation of reputation data
- **R2**. Local computation of trust and risk values
- **R3**. Dynamic decision making based on trust, reputation and risk

In addition, since Post-Federation focuses on making decisions about granting/revoking privileges, or even terminating the federation, additional requirements are to be considered:

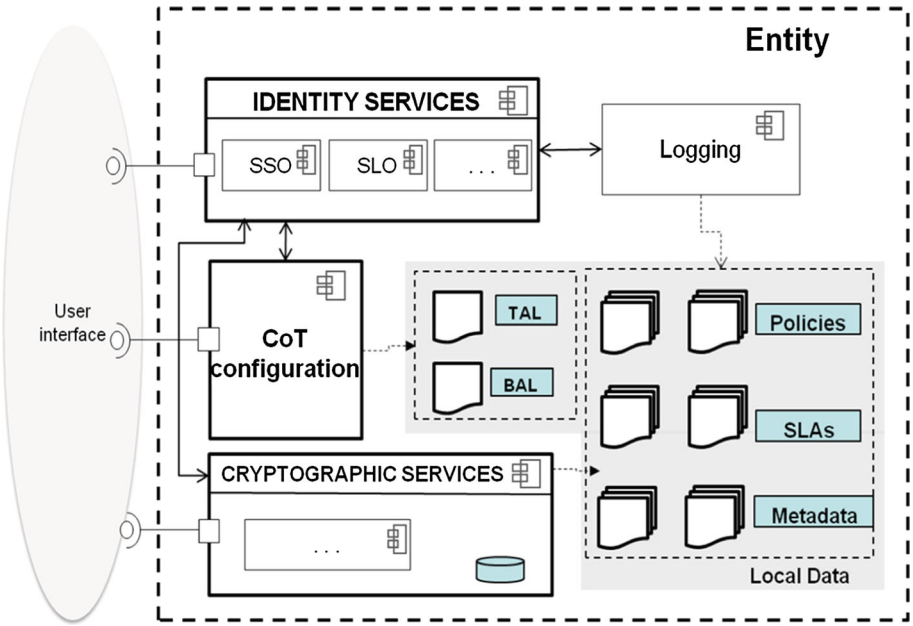


Fig. 1 Generic FIM architecture based on static agreements

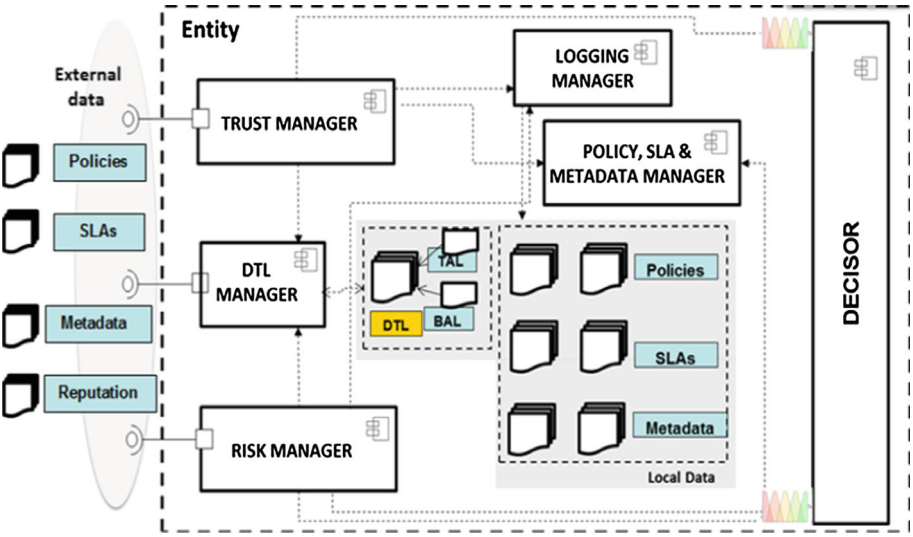


Fig. 2 Architecture for entities in the reputation-based federation model

– **R4**. Monitoring and adjustment of trust levels: transaction history, trust update, Service Level Agreement (SLA) conformance, etc. Based on this set of requirements, Fig. 2 shows the proposed extended architecture that is to be implemented by entities taking part in the reputation-based federation model, namely IdPs and SPs.

In the proposed architecture, the initial cryptographic and identity services modules remain unchanged, so we do not show them in Fig. 2 for better clarity. The main modification consists on breaking the CoT configuration component into several blocks that allow to fulfill the requirements to achieve dynamism. Next, we explain the details of every element in the architecture:

1. **Local Data:**
   - **Dynamic Trust List (DTL)** stores trust and reputation data regarding other entities in the FIM infrastructure. It is automatically updated according to the establishment and evolution of trust relationships. Furthermore, to maintain compatibility with existing deployments and allow the establishment of relationships based on previous agreements, Trust Anchor Lists (TALs) and Business Anchor Lists (BALs) can also exist and, when this is the case, they will be used to initialize the DTL. The TAL contains the set of entities and associated keys that are trusted for authentication purposes; whereas the BAL, stores the list of entities with which direct business trust relationships have been established. These two lists are used in current SAML federations and its usage is better detailed in the specifications [9].
   - **Policies, SLAs and Metadata** Entities define policies regarding different aspects of FIM, such as the supported cryptographic algorithms, thresholds and rules for risk, trust and reputation values, privacy-related rules, etc. Besides the policies, each entity defines SLAs to describe the extent of federation relationships, e.g., type of identity attributes to be exchanged.

   Finally, each entity has Metadata to specify the technical information required to configure a federation relationship using a particular framework (e.g., SAML Metadata [21]): supported bindings and use-cases, digital certificates for secure communication, etc.

2. **External Data:**
   - **Policies, SLAs and Metadata** of the transacting entity required for comparison with local information in order to ensure that federation is technically and legally feasible to some desired extent.
   - **Reputation Data** will be requested in order to compute an initial trust level if the entity is unknown (i.e., not contained in the DTL).

3. **Managers:**
   - **Trust Manager** processes external and local trust information, executes the reputation protocol, and aggregates opinions and computes trust values.
   - **Risk Manager** evaluates the risk associated with the current transaction using data from the DTL Manager, the Policy, SLA and Metadata Manager, and external sources.
   - **DTL Manager** handles DTL CRUD (create, read, update and delete) operations. Essentially, performs the operations required by the Risk and Trust Engines. The list is dynamically updated under specific events triggered by the Trust Engine, e.g. when a successful interaction ends. It also reads trust values, and time related information to compute the risk associated with particular transactions.
   - **Policy, SLA and Metadata Manager** communicates trust related rules to the Trust Engine (e.g. thresholds for malicious entities, default trust values, etc.), provides input information to the Risk Engine, performs CRUD operations over the local data (e.g., if a trust relationship evolves positively, the Trust Engine will notify this module

Table 1 Reputation-based federation model cases

	Case (A)	Case (B)	Case (C)	Case (D)
Entity in DTL	No	No	Yes	Yes
Entity in federation	No	Yes	No	Yes

to extend the initial SLA and grant more permissions to the transacting entity), and provides policy information to the Decisor.

4. **Decisor** decides whether to initiate/accept or not a transaction with another entity. The inputs for this module are the computed trust value, the risk associated with the transaction and the local policies that will be used to govern the decisions.

2.2 To Federate or Not To Federate: Making Dynamic Decisions

Depending on the information contained in the DTLs of the transacting entities and the federations where they belong, we envision four possible cases under the reputation-based federation model (see Table 1).

All cases are triggered when a SP wants to transact with an IdP, driven by a user requesting a service (see Fig. 3). The goal of our model is to move from cases (A) or (B), to case (D) and maintain relationships in this last scenario if the behavior of the involved entities is good. When the behavior is bad or the federation has to be terminated for other reasons, the relationship moves to the state in case D. The transition (A)→(D) implies the establishment of both a federation and a trust relationship, while the transition (B)→(D) only requires the establishment of an initial trust value.

- **Case (A):** Newcomer entering a federation. In an existing federation where the IdP belongs, a new entity (SP) wants to join and become a member. Since each entity is unknown to the other (no previous interaction exists), there is no information in the DTLs.
- **Case (B):** Transaction between unknown entities in a federation. Both the SP and the IdP belong to the same federation, i.e., they agreed on common policies and technologies to use. Yet, there is no previous interaction between them, so not trust information is available in their DTLs.
- **Case (C):** Transaction request from a de-federated provider. Previous information regarding the requesting provider exists in the DTL but it is not federated because whether the associated trust value was not enough to continue in the federation, or because the federation was finished for other reasons. When receiving this kind of request, the entity can chose among using the stored trust value or asking for fresh reputation data.
- **Case (D):** Transaction between known entities in a federation. Similar to case (B) but here the SP and IdP know each other from previous interactions, so trust information is available in their DTLs.

The flowchart summarizing the operation of the proposed dynamic federation model is shown in Fig. 3. Basically, the first step when a request is received from an unknown provider is to check if it is contained in the DTL. If the entity is not in the DTL, then the IdMRep protocol needs to be executed to obtain reputation data about the unknown provider. Next, in case the trust value based on reputation is acceptable, the appropriate request is sent depending on if the entity is already federated or not. In case the trust value is not enough for cooperation, then the transaction is avoided.

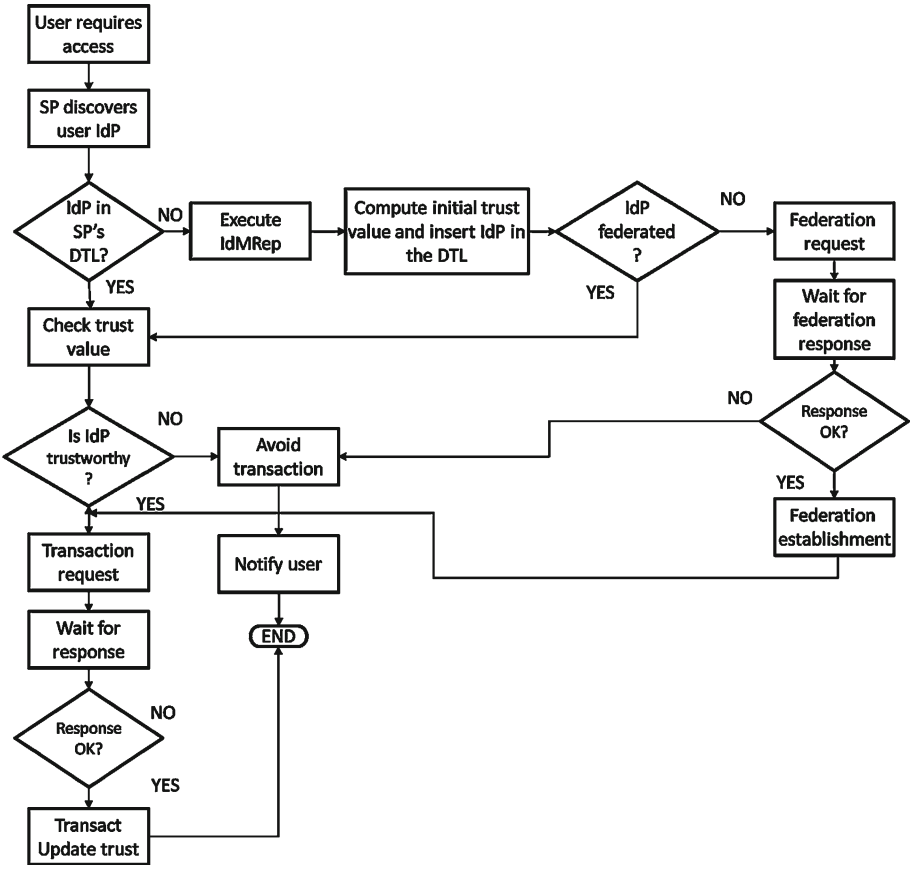


Fig. 3 Flowchart for dynamic federation

3 IdMRep: A Reputation Model for IdM

3.1 Network Model, Dissemination and Storage

To achieve our goal of shifting from static pre-configuration to a peer-to-peer (P2P) behavior when establishing federation relationships, we define a new trust logic overlay called "*unstructured P2P based on DTL*". If a participating entity has a DTL entry for a specific entity in the FIM network, then there is a directed edge from the former to the latter. On bootstrapping, DTLs are initialized based on the pre-configured trust relationships and agreements existing in each entitys TAL or BAL, and so the overlay is created based on these data. Then, the overlay will dynamically change according to the current state of the trust relationships based on entities behavior, and on the entities joining/leaving the system. Over this DTL-based unstructured P2P model, we define a new protocol, IdMRep, allowing us to gather reputation data.

In order to gather reputation data about unknown entities we define a dissemination approach called "Query Flooding based on DTL". The messages involved in this approach are two: `ReputationRequest` and `ReputationResponse`.

A `ReputationRequest` message is used to ask for reputation and it contains the following fields:

- **Message ID:** Message identifier
- **Reputation Requester:** The entity (SP/IdP) asking for reputation data
- **Subject:** The subject of the reputation or **"reputee"**, i.e., the entity (SP/IdP) whose reputation score is being calculated
- **Time to Live (TTL):** Number of times the `ReputationRequest` is forwarded through the FIM network. This number imposes a hop limit that constrains the lifetime of the request in the network. The TTL associated to a message starts with a 0 value and is increased after each forwarding until it reaches its limit, which is the value specified in this field.
- **Context (Cx):** Reputation is associated to a context (e.g., *"making good authentication assertions"*, *"maintaining privacy"*, etc.), so this field is used to specify the list of contexts for which the requester wants to get reputation data about the subject

In turn, `ReputationResponse` messages are used to convey reputation data in reply to a `ReputationRequest`, and they contain the following fields:

- **Message ID:** Message identifier
- **ReputationResponder:** The entity sending the reputation data
- **Timestamp:** Time when the reputation data was calculated
- **Reputation data:** Associated to Subject, regarding Contexts Cx

The above messages to gather reputation data by applying the following logic operation:

1. If the entity chosen for a transaction is unknown, then a `ReputationRequest` message is sent to the entities in the DTL.
2. When receiving a `ReputationRequest` message, the entity must check if there is an entry for the *reputee* in its DTL. If so, it must construct and send a `ReputationResponse` message back to the requester (experiments with malicious behaving entities will be addressed ion Sect. 4).
3. An entity should forward incoming `ReputationRequest` messages to the entities in its DTL, except to the one that delivered the incoming query.
4. An entity receiving a message with the same Message ID and *reputee* as another one received before, must discard such message.
5. After receiving all the `ReputationResponse` messages, the original requesting entity must aggregate them to obtain a final global reputation value (see Sect. 3.2 for details on the aggregation procedure). The *reputee* is then added to the DTL with the initial computed trust value. If there are no responses, the requesting entity can choose between adding the entity with an initial positive default trust value or not adding the entity to the DTL.

In Fig. 4 we show the protocol sequence diagram for a particular example transaction in a FIM network with five providers (SP1, SP2, SP3, IdP1 and IdP2) and the following information in their DTLs (Table 2):

More specifically, SP1 wants to initiate a transaction with IdP2, which is not present in its DTL.

The DTL constitutes the key element to maintain trust related information and its content is updated on-the-fly as the FIM system evolves, instead of being filled statically previous to interaction. Basically, every entry in the DTL contains:

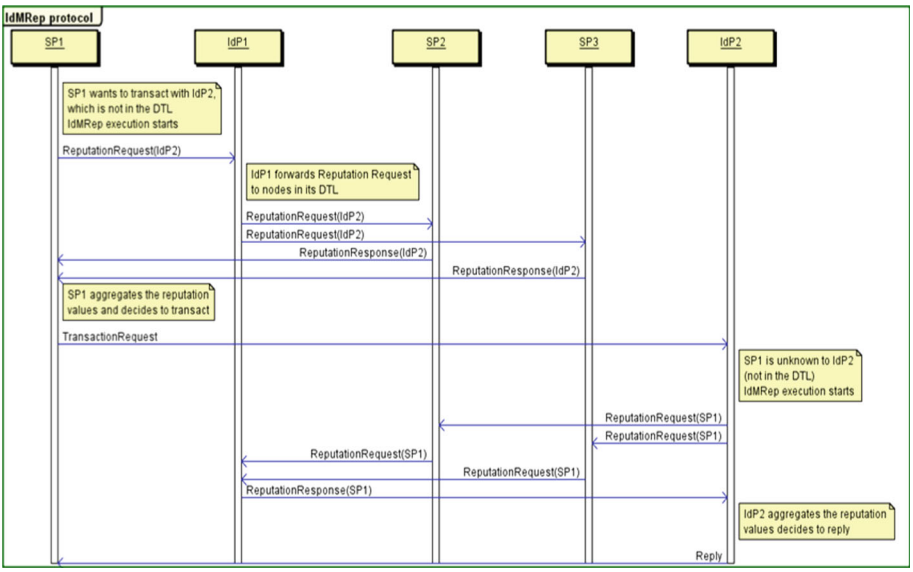


Fig. 4 Flowchart for dynamic federation

Table 2 Sample FIM network

Provider name	DTL content
SP1	IdP1
SP2	IdP1, IdP2
SP3	IdP1, IdP2
IdP1	SP1, SP2, SP3
IdP2	SP2, SP3

- **Entity ID:** identifier of the entity whose trust and reputation data are stored.
- **Reputation Data:** information required to compute the reputation value associated with the entity with identifier Entity ID. It depends on the applied reputation computation function.
- **Trust Data:** information required to calculate the local trust value associated with the entity with identifier Entity ID. It depends on the applied trust computation function. Furthermore, other trust material such as the digital certificates required for secure communication, are stored here.

3.2 Reputation and Trust Computation

To make dynamic trust decisions in FIM networks, reputation about the unknown entities must be first computed based on the received reputation responses; next, an initial trust value must be computed based on this reputation.

For such purpose different formulas can be used and so our architecture accepts any computation mechanism to be plugged in the Trust Manager. Here, we use formulas (1), (2) and (3) to calculate the local trust values and to aggregate the reputation opinions, respectively. We have chosen these formulas because they are easy to understand and implement, lead to good results and illustrate the operation of the proposed reputation-based federation model.

However, more complex functions may be used, e.g., considering timing and data freshness issues or different reputation contexts.

$$Ttot^{(0)} = \begin{cases} Td \rightarrow & \textit{initial trust when preconfigured relationship exists} \\ Tr \rightarrow & \textit{reputation value when no preconfigured trust exists} \end{cases} \quad (1)$$

$$Ttot^{(n)} = \alpha \cdot Ttot^{(n-1)} + (1-\alpha) \cdot Sat^{(n)} \quad (2)$$

where α, $\text{Ttot}^{(n)}$, $\text{Sat}^{(n)}$, Td, Tr $\in$ [0,1]. Ttot is the total trust value assigned to an entity based on the existing evidences and default values; $\text{Sat}^{(n)}$ is the satisfaction value of the nth transaction (equal to 1 in case the transaction is satisfactory and 0 otherwise); α is an adjustable parameter used for tuning the importance of recent and older transactions; and the reputation value Tr is calculated as shown in Eq.(3):

$$Tr^{(m)} = \begin{cases} d & \text{when } m = 0 \\ r^{(m)} & \text{when } m = 1 \\ \beta \cdot Tr^{(m-1)} + (1-\beta) \cdot r^{(m)} & \text{when } m > 1 \end{cases} \quad (3)$$

where $\beta = 0.5$ and $\text{r}^{(m)}$, Tr $\in$ [0,1]. Tdefault is the default trust value assigned when no reputation data is obtained after executing IdMRep, $\text{r}^{(m)}$ is the value of the m"th" reputation vote. It is to say that the reputation value sent in a `ReputationResponse` message is the local trust value that the responding entity has in its DTL for the Subject. Besides the aggregation functions, we also need to define the decision policies in order to complete the model. Again, although more complex policies may be applied, we chose a simple trust policy with the following rule "**IF** Ttot $\geq$ MALTHRESHOLD **THEN** transact", where MALTHRESHOLD is a threshold determining the minimum trust value required to transact with another entity.

4 Evaluation

4.1 Simulations Description and Setup

This section is dedicated to the validation of the proposed model in order to prove the benefits of including reputation data in FIM networks. This aspect is complex to test in a real world scenario since it is would imply the deployment of a complex infrastructure with a high number of providers, which is not a trivial task. Thus, we have opted for a simulation-based validation using OMNeT++ [22], which is an open source C++ simulation framework widely used in the scientific community.

We differentiate between two types of FIM networks: networks with relationships between SPs and IdPs and networks with SP-IdP and IdP–IdP relationships. A relationship between providers means that they can transact with each other following a federation protocol. In summary, Table 3 compiles all the parameters used in our simulations. For the sake of simplicity, we consider that only SPs can act maliciously.

The chosen number of providers is based on a public survey[1] regarding the composition of deployed federations. According to this survey, the average values for providers are around 25 SPs and a lower number of IdPs. Though there are federations with a higher number of entities, we leave the modeling of bigger networks for future experiments. It is to say that the number of repetitions, though small, allows us to obtain confidence intervals of 95 % since the variance in the measurements between experiments was very low.

[1] http://kantarainitiative.org/confluence/display/concordia/Identity+Federation+Survey+Results [Available online; last accessed April 2013].

Table 3 Simulation parameters

FIM network model	
#IdPs	Number of identity providers: 5, 10, 15, 20
#SPs	Number of service providers: 25
ConnectivitySI	Connectivity degree between SPs and IdPs, (preexisting relationships): 0.2
ConnectivityII	Connectivity degree between IdPs and IdPs (preexisting relationships): 0.075
Network type	Connections between SPs-IdPs (A) or connections between SPs-IdPs and IdPs-IdPs (B)
Trust and reputation model	
Td	Initial trust for pre-existing relationships: 1
Tdefault	Initial default trust value assigned to an entity when no reputation data Tr is available: 0, 0.5
α	Parameter in the trust model to adjust the importance of new transactions and old transactions: 0.5
MALTHRESHOLD	Trust threshold for decision making (i.e., if local trust for entity i is greater or equal than MALTHRESHOLD, then a transaction can be initiated/accepted): 0.5
Entity behavior model	
MalRate	Fraction of malicious entities in the network: 0, 0.1, 0.2, 0.3, 0.4, 0.5
IdMRep forwarding model simulation	
TTL	Time to live for reputation requests: 1, 2, 3, 4, 5, 6
#repetitions	Number of repetitions per experiment: 5

The operation for each simulation running implies that nodes randomly chose another node in the network to transact with and execute the IdMRep protocol if there is no entry in the DTL for the chosen node. This behavior is constantly repeated through the duration of the experiment. The simulation time is long enough to cover a high number of interactions (250 requests are sent by SPs in average).

We introduce different fractions of malicious entities to analyze their impact on the accuracy. Each experiment is repeated 5 times with varying random seeds. The seed influences the order in the sequence of initiated transactions.

Additionally, we consider the following metrics:

- The **message overhead**. We look at the overhead caused by the extra messages issued when using the reputation protocol. The message overhead for a node i (MO_i) in a FIM network is calculated as shown in Eq. (4):

$$MO_i = \frac{RepRequest_i + RepResp_i}{IntendedTransactions_i} \tag{4}$$

  where $RepRequest_i$ and $RepResp_i$ refer to the number of `ReputationRequest` and `ReputationResponse` messages sent by node i, and $IntendedTransactions_i$ is the number of transactions initiated by the node.
- The **accuracy** (or success rate), which reflects the percentage of successful transactions. We calculate the success rate for a node i (SR_i) as the number of interactions with good entities plus the avoided interactions with malicious entities over the total number of transactions performed by the node, as shown in Eq. (5):

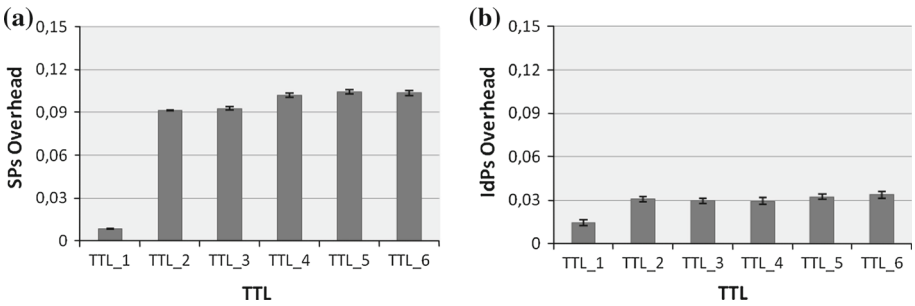


Fig. 5 Average overhead for SPs (**a**), and for IdPs (**b**) when varying the TTL parameter, d = 0.5

$$SR_i = \frac{TransGood_i + AvoidedTransMal_i}{TotalTransactions_i} \tag{5}$$

where $TransGood_i$ is the number of transactions accepted by node i coming from a good entity; $AvoidedTransMal_i$ is the number transactions rejected by node i and coming from malicious entities; and $TotalTransactions_i$ is the total number of transactions in which node i can participate during the running of a simulation.

4.2 Simulation Results

4.3 Outcomes Under "Nice" Conditions

We first analyze the protocol supposing "nice" conditions in the simulated environment, i.e., without considering malicious nodes (MalRate = 0). For this analysis we use the parameters in Table 3 and observe the behavior of the protocol in terms of overhead and accuracy with varying TTL. For simplicity, we only analyze networks of type B with 5 IdPs. Here, we refer to the accuracy metric as the "transaction rate" because when there are no malicious entities, all the completed transactions will be successful.

We conducted two opposed experiments with the aim to show the benefits of IdMRep with regards to the current FIM frameworks. Results for both experiments are graphically shown in Figs. 5, 6. In the first experiment entities are willing to cooperate even if no reputation data are found about other unknown providers (i.e., Tdefault = 0.5); whereas in the second experiment entities only transact with other unknown entities if reputation data are available (i.e., Tdefault = 0).

The first experiment shows the benefits and disadvantages of applying IdMRep to the "*accept-all-comers*" model. The average transaction rate is 100 % since entities are always trustworthy. Thus, in a nice environment may seem useless to introduce reputation, but the results in the next section will show the benefits of the approach when there are malicious entities. Also, in a "nice" environment, different good reputation values may help to decide on giving or denying different types of transactions.

Similarly, with the second experiment (Figs. 7, 8) we tested the benefits and disadvantages of introducing the reputation protocol in a rigid FIM model. The average transaction rate in this case is not 100 %, since now entities do not trust other entities if there is no reputation available. Instead, the transaction rate value is around 60 % for all TTL cases, which drastically improves the case of using a rigid trust model where the percentage of transactions with

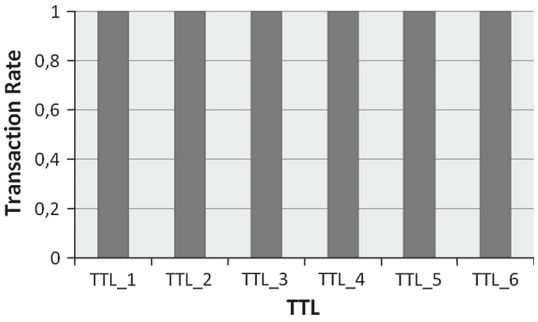


Fig. 6 Average transaction rate when varying the TTL parameter, Tdefault = 0.5

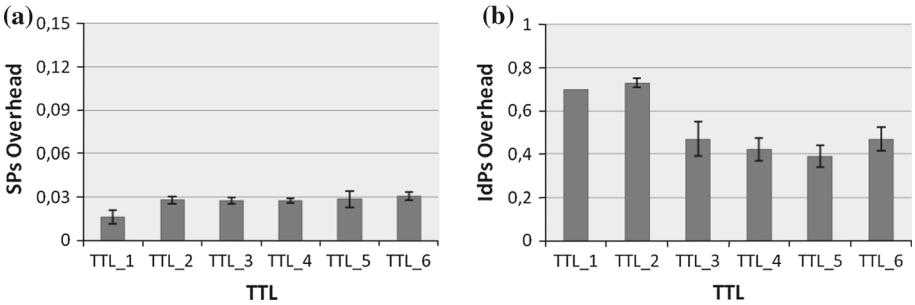


Fig. 7 Average overhead for SPs (**a**), and for IdPs (**b**) when varying the TTL parameter, Tdefault = 0

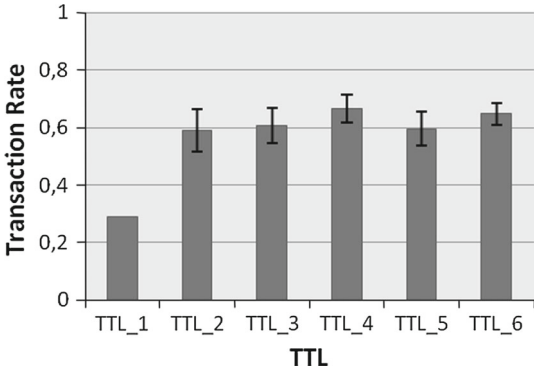


Fig. 8 Average transaction rate when varying the TTL parameter, Tdefault = 0

unknown entities is 0 %. As a negative counterpart, the average overhead for SPs is a bit higher than the obtained in the case of using Tdefault = 0.5.

In any case, the overhead is reasonably low to positively consider the introduction of the protocol in FIM networks. Furthermore, the TTL value does not influence significantly the overhead. This behavior will be also observed later.

As we can see, the proposed protocol allows us to achieve a trade-of between the trust-all-comers and the rigid trust model, imposing a reasonably good overhead.

In Fig. 9 we show the accuracy obtained in a network of type A, where the Tdefault value assigned to unknown entities is 0.5. The number of IdPs is increased from an initial value of 5 to a final value of 20 according to Table 3, these two extreme cases are the ones shown in

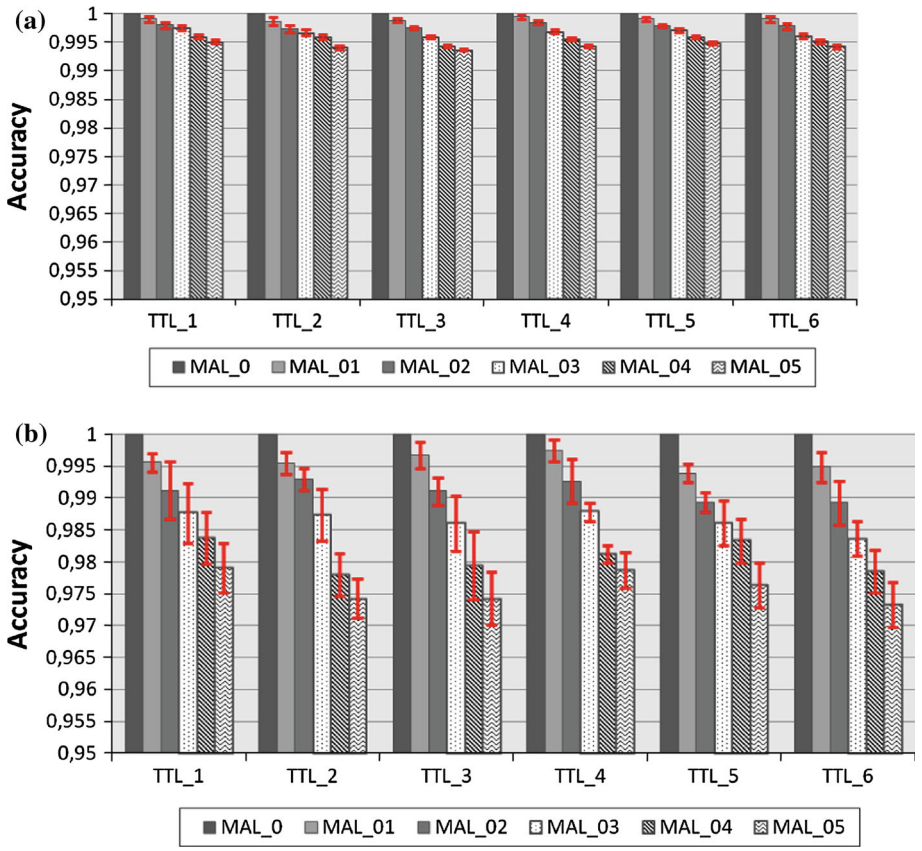


Fig. 9 Accuracy measured in Network type A with parameter Tdefault = 0.5 varying TTL, MalRate and #IdPs. **a** Accuracy for increasing percentage of malicious nodes varying the TTL, Network type A, Tdefault = 0.5, #IdPS = 5. **b** Accuracy for increasing percentage of malicious nodes varying the TTL, Network type A, Tdefault = 0.5, #IdPS = 20

the figure. Besides, the results are presented for TTL values between 1 and 6, and percentage of malicious nodes ranging from 0 to 50 %.

Having 5 IdPs, the obtained accuracy is always over 99 % and it decreases with the percentage of malicious entities. As it can be observed, increasing the TTL value does not have a significant impact on the accuracy. The same tendency regarding malicious behavior and TTL variations is observed in the rest of the tests for networks with 10, 15 and 20 IdPs, so we will not show more graphs with the evolution of the TTL parameter. On the other hand, the accuracy gets lower when the number of IdPs increases: around 98 % for 10 IdPS, 97.5 % for 15 IdPs and 97 % when the number of IdPs reaches 20.

Since the TTL does not have a significative impact, the next graph in Fig. 10 represents the accuracy depending on the kind of nodes in the network. It can be seen that the accuracy is slightly better when the entities are conservative, i.e., Tdefault = 0. By splitting the accuracy in four components: (1) accepted transactions from reputable entities, (2) accepted transactions from non-reputable entities, (3) rejected transactions from reputable entities, and (4) rejected transactions from non-reputable entities; we observed that the number of rejected transactions coming from reputable entities is higher in the case of having conservative entities.

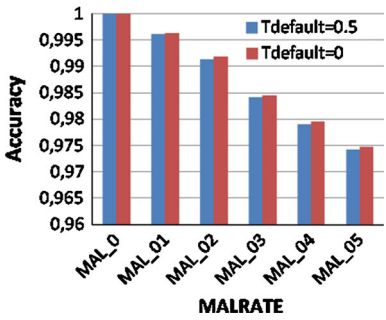


Fig. 10 Accuracy for different rates of malicious entities, Tdefault = 0, 0.5

5 Background Technologies and Related Work

SAML (Security Assertion Markup Language) version 2 from OASIS [3] along with ID-FF (Identity Federation Framework) [6] from Liberty Alliance constitute current deployed FIM standards. They define an XML based framework to allow the exchange of assertions (i.e., authentication, attributes, and authorization) between providers. Such providers use Metadata exchange for establishing federations. WS-Federation (Web Services Federation Language) version 1.2 [7], in turn, is another approach originated from the industry, which has been standardized within OASIS. WS-Federation allows security realms to federate, focusing on Web Services environment. OpenID [4] is a more limited specification, which provides only user-centric Web SSO. The OAuth protocol [8] has also emerged as an implicit SSO protocol, similar to OpenID. Its aim is to delegate authorization from web pages to a central authority. Scientific work related to dynamic management of federations is scarce. PrivaKERB [23] efficiently tackles the underlying problems of user anonymity and service access untraceability within the Kerberos protocol, even in cross-realm transactions where users roam away from their home network. Yet, it is still assumed both the end user and service provider to have a pre-established trust relationship with the Key Distribution Center (KDC) [24] presents a reputation-based trust management method that uses a single Trusted Third Party (TTP) for aggregating the IdP vote and transmitting the result to the SPs. This approach could have scalability problems since it requires to pre-configure relationships with the TTPs, a process that grows in complexity when a high number of entities exist in the system [25] and [26] propose integrating trust negotiation into IdM systems [25] includes bilateral credential disclosure techniques between SPs and between users and SPs in federated IdM, called FAMTN (Federated Attribute Management and Trust Negotiation), while [26] proposes negotiation based on matching of release policies into InfoCards, called identity meta-system. These approaches are mainly focused on flexible access control but do not deal with the establishment of dynamic trust relationships between providers. In this sense, the proposals that are closest to the ideas presented in this paper are [18,19] and [17]. However, thought they introduce reputation in FIM, they all require some form of centralized storage; so the move to a completely distributed model is the main novelty of our approach.

In regard to our work, we have been researching towards the definition of a trust and decision model for dynamic identity federation that combines both reputation and risk evidences. In [20,27] we set the basis to extend the generic trust management model called PTM (Pervasive Trust Management) [28] in order to include risk management, reputation, and SLA negotiation. Based on this, our work in [29] defines the risk computation part of the architecture, while here we focus on the reputation part. In this sense, we observed that

the integration of risk evaluation with trust management is gaining attention in the context of dynamic federation [14].

6 Conclusions and Future Work

Current FIM frameworks are not suitable for dynamic open environments, such as wireless communication networks. Specifically, the underlying trust models are poorly defined or too rigid to allow an agile and secure way of establishing relationships. To solve this problem, we have presented a reputation-based federation model, which includes the definition of a network model and a new protocol to gather reputation data. Furthermore, the presented model can be implemented over any of the existing federation protocols. Evaluation results show the feasibility of the solution, since the obtained performance from the point of view of both accuracy and overhead is good. Besides, the model allows the establishment of new relationships between previously unknown entities guaranteeing a high success rate. To this end, we are currently investigating how to improve and adapt the initial version of the reputation protocol in order to tackle more complex malicious patterns. As immediate future lines, we plan to study the effect of changing the forwarding policy used by IdMRep, analyze the impact of different trust engines, enrich the trust and reputation formulas to capture context information, and implement the risk engine to make better decisions.

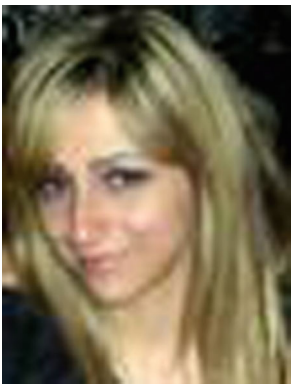
Acknowledgments The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of this paper.

References

1. Maler, E., & Reed, D. (2008). The venn of identity: Options and issues in federated identity management. *IEEE Security & Privacy*, *6*(2), 16–23.
2. Chadwick, D. W. (2009). Federated identity management. In: A. Aldini, G. Barthe & R. Gorrieri (Eds.), *Foundations of security analysis and design* (pp. 96–120). Berlin: Springer.
3. Ragouzis, N., Hughes, J., Philpott, R., Maler, E., Madsen, P., & Scavo, T. (Eds). (2008). *Security assertion markup language (SAML) V. 2.0. Technical overview*. OASIS Comittee Draft 02.
4. OpenID specification. http://openid.net/developers/specs/. Accessed April 2013.
5. Liberty Alliance Initiative. http://projectliberty.org Accessed February April 2013.
6. Wason, T. (Ed.), (2009). *Liberty ID-FF architecture overview, version 1.2*. Liberty Alliance Project.
7. Nadalin, A., & Kaler, C., (Eds.) (2006). *Web Services Federation Language (WS-Federation), version 1.1*.
8. Recordon, D., & Hardt, D. (2012). *The OAuth 2.0 authorization protocol*. IETF Network Working Group.
9. Boeyen, S., Ellison, G., Karhuluoma, N., MacGregor, W., Madsen, P., Sengodan, S., Shinjar, S., & Thompson, P., (2003). *Liberty trust models guidelines*. Liberty Alliance Project, version 1.0.
10. Jensen, J. (2011). 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* (pp. 1–12). Springer.
11. Smith, D. (2008). The challenge of federated identity management. *Elsevier Network Security*, *4*, 7–9.
12. Landau, S., Le Van Gong, H., & Wilton, R. (2009). Achieving privacy in a federated identity management system. In R. Dingledine & P. Golle (Eds.), *Financial cryptography and data security* (pp 51–70). Berlin: Springer.
13. Arias Cabarcos, P., Almenárez Mendoza, F., Andres Marín Lopez, A., & Díaz Sanchez, D. (2009). Enabling SAML for dynamic identity federation management. In J. Wozniak, J. Konorski, R. Katulski, & A. R. Pach (Eds.), *Wireless and mobile networking* (pp. 173–184), Berlin: Springer.
14. ETSI GS INS-004 V 1.1.1, Group Specification. (2011). Identity and access management for networks and services; Dynamic federation negotiation and trust management in IdM systems.

15. Josang, A., Ismail, R., & Boyd, C. (2007). A survey of trust and reputation systems for online service provision. *Decision Support Systems*, *43*(2), 618–644.
16. Gómez Mármol, F., & Martínez Pérez, G. (2010). Towards pre-standardization of trust and reputation models for distributed and heterogeneous systems. *Computer Standards & Interfaces*, *32*(4), 185–196.
17. Gómez Mármol, F., & Girao, J. (2010). TRIMS, a privacy-aware trust and reputation model for identity management systems. *Computer Networks*, *54*(16), 2899–2912.
18. Boursas, L., & Danciu, V. A. (2008). Dynamic interorganizational cooperation setup in circle-of-trust environments. In *IEEE network operations and management symposium*. (pp. 113–120).
19. Xiang, Y., Kennedy, J. A., Richter, H., & Egger, M. (2010). Network and trust model for dynamic federation. In *The fourth international conference on advanced engineering computing and applications in sciences*, (pp 1–6).
20. Arias Cabarcos, P. (2011). Risk assessment for better identity management in pervasive environments. In *IEEE international conference on pervasive computing and communications workshops (PERCOM workshops)* (pp. 389–390).
21. Cantor, S., Moreh, J., Philpott, R., & Maler, E. (Eds.) (2005). *Metadata for the OASIS security assertion markup language (SAML), V2.0. OASIS standard*.
22. OMNeT++. http://omnetpp.org. Accessed April 2013.
23. Pereniguez, F., Marín-López, R., Kambourakis, G., Gritzalis, S., & Gómez, A. F. (2011). PrivaKERB: A user privacy framework for Kerberos. *Computers & Security*, *30*(6), 446–463.
24. Choi, D., Jin, S. H., & Yoon, H. (2007). Trust management for user-centric identity management on the internet. In *Proceedings of IEEE international symposium on consumer electronics* (pp. 1–4). Dallas, TX, USA.
25. Bhargav-Spantzel, A., Squicciarini, A. C., & Bertino, E. (2007). Trust negotiation in identity management. *IEEE Security & Privacy, 5*(2), 55–63.
26. Abliz, M., (2009). *Negotiating trust in identity metasystem*. University of Pittsburgh Department of Computer Science, Technical Report. TR-10-173.
27. Almenárez, F., Arias, P., Marín, A., & Díaz, D. (2009). Towards dynamic trust establishment for identity federation. In *Proceedings of the ACM Euro American conference on telematics and information systems*.
28. Almenárez, F., Marín, A., Díaz, D., Cortés, A., Campo, C., & García, C. (2011). Trust management for multimedia P2P applications in autonomic networking. *Ad Hoc Networks, 9*(4), 687–697.
29. Arias-Cabarcos, P., Almenárez-Mendoza, F., Marín-López, A., Díaz-Sánchez, D., & Sánchez-Guerrero, R. (2012). A metric-based approach to assess risk for "On Cloud" federated identity management. *Journal of Network and Systems Management*, *20*(4), 513–533.

Author Biographies



Patricia Arias Cabarcos received her Telecommunication Engineering degree from University Carlos III of Madrid (UC3M) in 2008. She obtained the MSc and Ph.D. in Telematics Engineering from UC3M in 2009 and 2013, respectively. She is currently working as researcher within the Pervasive Computing research group. Her research focuses on the problem of identity management in open and dynamic environments, with special attention to risk analysis and the underlying trust models.

Florina Almenárez received her Computer Science degree in 1999 from University Autónoma of Bucaramanga, and the Ph.D. degree from the University Carlos III of Madrid in 2006. She is currently an associate professor at UC3M. She received an award-winning as Magna CumLaude in her Computer Engineering degree. Her research interests include trust management and reputation management models, identity federation, security in ubiquitous computing, and SIM-based applications. She leads the research activities of the PerLab group in advanced trust models, security architectures for open and dynamic spaces, and identity management.

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 an M.Sc. and Ph.D. in computer engineering from the University of Murcia. Contact him at felix.gomez-marmol@neclab.eu

Andrés Marín received a Telecommunication Enginering degree and Ph.D. from the Technical University of Madrid in 1992 and 1996 respectively. He lectures in Computer Networks and Ubiquitous Computing in the University Carlos III of Madrid, as an associate professor. His research interests include ubiquitous computing: limited devices, trust, security services, and security in NGN.