D-FOAF: Distributed Identity Management with Access Rights Delegation

Sebastian Ryszard Kruk[1], Sławomir Grzonkowski[1,2], Adam Gzella[1,2], Tomasz Woroniecki[1,2], and Hee-Chul Choi[3]

[1] Digital Enterprise Research Institute, National University of Ireland, Galway, Ireland[*]
<firstname.lastname>@deri.org

[2] Faculty of Electronics, Telecommunications and Informatics, Gdansk University of Technology, Poland

[3] DERI Seoul National University, 28-22 Yeonkun-Dong, Chongno-Ku, Seoul 110-749, Korea

Abstract. Todays WWW consists of more than just information. The WWW provides a large number of services, which often require identification of it's users. This has lead to the fact that today users have to maintain a large number of different credentials for different websites - distributed or shared identification system are not widely deployed. Furthermore current authorisation systems requires strict centralisation of the authorisation procedure - users themselves are usually not enabled to authorise their trusted friends to access services, although often this would be beneficial for services and businesses on the Web.
In this article we present D-FOAF, a distributed identity management system which deploys social networks. We show how information inherent in social networks can be utilised to provide community driven access rights delegation and we analyse algorithms for managing distributed identity, authorisation and access rights checking. Finally we show how the social networking information can be protected in a distributed environment.

1 Introduction

The Internet provides a large number of different services. Usually services require authentication of their customers. Two most common examples that require user authentication are access control to services or resources, and personalisation of aservices. The usuability of services suffers greatly from the fact the usually no single sign in facility is available.

[*] The support of Enterprise Ireland, through its Informatics Commercialisation initiative, for the eLITE Industry Led Research Project in eLearning is gratefully acknowledged. This material is also partially based upon works supported by KBN, Poland under grant No. 4T11C00525, and partially by a grant of the Interoperable EHR Research and Development Center(A050909), Ministry of Health & Welfare, Republic of Korea. The authors would like to acknowledge Stefan Decker, Marco Neumann, John Breslin, the DERI Semantic Web Cluster and the Corrib.org working group for fruitfull discussions.

The proliferation of Internet services introduces many problems like no single identity for Internet users or no scalability in trust and access rights management. Some of those problems has been so far addressed in a number of ongoing projects.

The main difference between the Internet and real world services are authorisation procedures. In the real world each person has a single identity expressed with a number of credentials like ID card, passport or driving license. This allows real world service providers to easily confirm the authenticity of the presented credentials In the Internet, each user has to deal with a number of identities with different credentials like login-password pair. Since the is no notion of single identity service providers are usually inclined to introduce new credentials for each user. As a result the trust to each user is build within each service separately.

Approaches like Microsoft Pasport [?], Sxip [?] or Liberty Alliance Project [?] are aiming to provides a solution to the single-sign-on problem. Due to various problems none of those projects has been widely adopted by service providers so far, making them useless for the majority of Internet users with the ever growing number of service.

Most of online services are usually based on very simple user profile management implementations that do not address problems stated above. Access rights are based on predefined, fixed list of groups and neither allow finer granularity nor trust delegation.

The notion of social networking emerged in the Internet with online community portals like Orkut that allow users to control access to the information based on the structure of the social network. Each user can restrict access to some parts of his/her profile information delegating trust within given number of degrees of separation.

1.1 Contribution and Paper Overview

In this paper we present an identity management solution based on social networks. Each user has a control on his/her profile and social networking information. Access rights are based on the structure of the social network and thus very fine grained by introduction of notion access rights delegation (see section **??**). Further on, we extend this solution to a distributed identity management system where only a single registration is required within the network of the user profile management systems (see section **??**). We present how the sensitive information from the perspective of the user and access rights management can be protected. Finally, we present the FOAFRealm system that implements presented solutions and utilises FOAF metadata to allow exchange of the profile information with other systems (see section **??**).

2 Use Case Scenario

One of possible scenarios where both distributed identity and the trust delegation is utilised is W3C information management. W3C consists of a growing number of member organisations. Each W3C Member has one Advisory Committee Representative (AC Rep). This person knows enough about the Member organisation's structure and forwards detailed technical reviews to the proper person. The AC Rep receives official notices from W3C. Acting as a gatekeeper, the AC Rep responds to, or delegates response to W3C Calls. The AC Rep appoints participants in W3C Working Groups.

Trust delegation When AC Rep has to grant access to some W3C services or resources, he/she needs to either add given person to an access control list or add this person to a group that already has access to the resource. In the constantly growing, evolving and changing research organisation managing access rights in that way maybe time consuming.

In this section we describe how AC Rep could delegate access rights without constant alteration of the ACLs or access groups.

AC Rep can define access rights group as a subgraph of social network within 2 degrees of separation from him/her. This allows his/her direct collaborators to delegate the access rights to W3C resources and services one step forward in the social network (see Fig. **??**). This way AC Rep does not have to alter the access rights list for every new member. It is enough when at least one of existing members establish friendship relation. The new member cannot delegate the access rights any further, though.

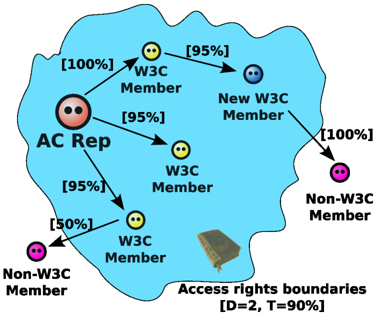

Fig. 1. W3C Scenario - Access rights delegation within the community

Many W3C Member organisations can take part in different W3C Working Groups. Access rights delegation based on the friendship relations may introduce security threats, by allowing people from different working groups to access resources allowed to other working groups. People affiliated with W3C Member can defined their friendship relations within *working group contexts*. But do not share access rights beyond working groups even though some of them stay in the direct friendship relation (see Fig. **??**).

Distributed identity management The identity management based on social networks provides a solution for fine granularity in access rights and trust delegation management. Some of the communities are spread across the number of different web applications, and many members very often belong to more then one. In our scenario, W3C consists of many independent member organisations that have their own identity management systems.

In today's world an typical user needs to remember a number of login-password credentials to access all of his/her accounts. When it comes to operate within the context of W3C people affiliated with member organisations have to manage additional account(s) to access the W3C resources and services. To ease the burden of handling multiple credentials and many friendship lists within different communities a distributed community driven profile management (see Def. **??**) can be established across a number of different web applications.

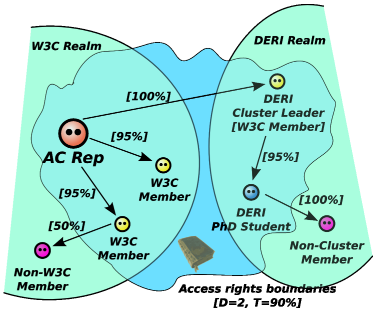

Fig. 2. W3C Scenario - Distributed identity management

Once all W3C Members agree on distributed identity management introducing new member organisations or new people affiliated with existing W3C Members will not force W3C to create and maintain new accounts. Additionally, there will be no need to add new access rights as they will be delegated into the local access control systems based on social networks.

3 Community Driven Access Rights Delegation

3.1 Social Networks as a Mean to Delegate Trust

In the contemporary Web 2.0 - full of wikis and community portals like Orkut [**?**] or LinkedIn [**?**] - wide community activity is perceived as a must for successful development of almost any Internet undertaking. By exploiting existing social networks to define access rights a system can easily evolve and eventually reflect the state of the real world.

Social networks driven identity management system (see Def. **??**) defines access rights in terms of friendship relations between users. Friendship relation can be naively modeled with a digraph, where a direct link from A to B means 'A knows B' [**?**].

Definition 1 (Community Driven Access Control). *The service S that implements identity management based on social networks UPM_{SN} provides the community driven access control over resources $\{r : r \in R_S\} \iff$ the changes introduced to the social network reflect the effective access rights $ACL(r)$ to the resource r.*

3.2 Going Beyond Friendship Digraph

The simple digraph representation does not cope with an important features presented in the real social networks – quality and context of friendship relation. To model social network more thoroughly each relationship can be annotated with metrics (see Def. **??**) defining how long the friendship lasts, frequency of meetings, average time spent together. For example Orkut [**?**] lets choose from few predefined levels of friendship (like *haven't met* or *good friend*). Though these examples give absolute and comparable numbers, they usually can not be used to measure user's real feeling about particular relationship. Smoothing each context scale (e.g. from 0% to 100%) helps to describe original connection more precisely.

Definition 2 (Friendship Level Metric). *Each friendship relation* $r \in R_{SN}$ *between social network member* $m_A \in M_{SN}$ *and member* $m_B \in M_{SN}$ *can be annotated with a quality measure* $FLM_{context}(m_A, m_B) \in < 0, 1 >$ *representing friendship level metric within given context.*

3.3 Calculating User Rights on the Social Network

Definition 3 (Social Networked Access Control List). *Access control list* $ACL_{SN}(m, d, l : m \in M_{SN}, d_{max} \in D_{SN}, flm_{min} \in FLM_{SN})$ *defined within user profile management system based on social networks defines access rights delegation within a maximal distance* $d_{max} \in D_{SN}$ *and a minimal friendship level metric* $flm\_context_{min} \in FLM_{SN}$*. Both values are computed across the social network SN from the one member* $m \in M_{SN}$ *to the member requesting access to the resource.*

One of primary functions of many web applications is to assure access rights control to particular resources defined with access control lists (see Def. **??**). Community driven access control system takes into account not only direct friends of the resource's owner but a whole social network. One of possible scenarios is when someone would feel that a very good friend of his/her very good friend is more trustworthy than a direct colleague he/she barely knows (see Fig. **??**).

A person is granted access to a resource when the friendship level and the distance between the resource owner and the service requester meet required constraints. Distance is the length of the shortest path from the owner to the requester. Final friendship level is computed by multiplying all metric values (which are all $\in < 0, 1 >$) on a path from resource owner to the requester (highest found product is taken). Access right can be delegated further only if other requesters conform to the given distance and friendship level constraints.

To find exact values of distance and friendship level a slightly modified Dijkstra algorithm [**?**] can be used. Although the Dijkstra algorithm has been proved to be quite efficient, operating on an enormous social network can introduce some scalability problems. However, finding the exact values of distance and computed friendship level is not required in the context of checking access rights. To check access rights to a resource the algorithm has to find whether the distance value is lower and the friendship level is higher than the given constrains. The modification made to the Dijkstra algorithm makes it stop calculations as soon as it finds 'yes or no' answer whether to grant access - without calculating precise values. Such an optimisation often saves a lot of time during the authorisation procedure.

4 Distributed Identity Management

The identity management based on social networks introduced in previous section (see Def. **??**) provides a solution for fine granularity in access rights and trust delegation management. Some of the communities are spread across the number of different web applications, and many members very often belong to more then one. To ease the burden of handling multiple credentials and many friendship lists within different communities a distributed community driven profile management (see Def. **??**) can be established across a number of different web applications.

Definition 4 (Distributed Community Driven Identity Management). *A federation of interlinked user profile management services based on social networks* $\{upm : upm \in UPM_{SN}\}$ *creates a distributed community driven identity management consisting of independent profile management services cooperating in authorisation and access rights calculating procedures.*

4.1 A Remedy for Multiple Accounts in the Federation of Services

User that has an account in one of the member services, called registration server [**?**,**?**] can easily log into the other member services of the distributed community driven identity management. The user has to remember one identity credentials representing his/her indentity while the system will perform distributed authorisation [**?**,**?**] algorithm (see Fig. **??**).

```
Require: $userName \neq null$ and $password \neq null$
Ensure: $auth_{result} \in \{true, false\}$
  $auth_D \Leftarrow$ perform local authentication
  if $auth_D \neq true$ then
    if $userServer \neq null$ then
      $auth_D \Leftarrow$ authenticate directly on user's server
    end if
    if $auth_D \neq true$ then
      $resultTable \Leftarrow$ perform query in network
      for $elem \in resultTable$ do
        if $elem[result] = true$ then
          $auth_D = true$
        end if
      end for
    end if
  end if
  return $auth_D$
```

Fig. 3. User authorisation in distributed environment

4.2 Protecting Social Network from Unauthorised Alterations

A social network and a distributed profile management system must be protected from many threats. The threats can be divided into several categories [**?**]

like human-related, cookies-related or fundamental problems. Unauthorised alterations of the profile information are one of the fundamental ones.

Access rights definition in community driven access management (see Def. **??**) is based on the structure of the social network. Therefore, social network information has to be especially protected by identity management system.

The improved security of the distributed social network [**?**,**?**] is introduced by signing local social networks (see Def. **??**) with the private key [**?**] issued by the registration server for the each user.

Definition 5 (Signature on the Local Social Network). *Each integral part of the social network $sn(m, s) \subset SN$ from the perspective of the member $m \in M_{SN}$ hosted by the service $s \in S_{SN}$ is accompanied with signature created with private key at the registration server $RS(m)$.*

The signature is checked every time the social network information is accessed. The registration server RS_{SN} is responsible for generating signatures for other federated services, protect the private key information and host the public keys.

4.3 Calculating User Rights on the Distributed Social Network

To allow user to access protected resources, the service has to check the presented credentials and confirm that the user conforms to the given access control list restrictions. In other words, the service has to check if distance and friendship level meet required constrains.

Distance and friendship level metrics computations are executed each time user wants or simply attempts to access the protected resource. The process of calculating user rights in distributed network is complex, and consist of three general steps:

Step 1 System utilises the modified Dijkstra algorithm [**?**] to compute distance (or friendship level) between the users. In the first step of the distributed computing, the algorithm is executed at the local service (see Fig. **??**). If local information conforms to the boundaries like maximal distance or minimal friendship level the algorithm terminates with success, otherwise it continues to the next step.

Step 2 In the second step, request is dispatched to each node and local computation is performed separately on each host in distributed social network. If any of the services can provide positive answer than the result is sent back to the service initiating the process and algorithm terminates.

Step 3 It might be assumed that close friendships are defined within one community managed by one of local authorisation services. It is also very possible that two people are connected through some other ones with their profiles on other nodes in the network. In this case, in order to compute distance between two people, system builds new digraph using information gathered from all hosts in network. When new digraph is created, Dijkstra algorithm is used to compute distance and friendship level metrics.

```
Require: $userA \neq null$ and $userB \neq null$ and $maxDist \in< 0, \infty >$ and
  $minLevel \in< 0, 1 >$
Ensure: $dist_{result} \in< -1, \infty >$ and $level_{result} \in \{-1\} \bigcap < 0, 1 >$
  $dist_D, level_D \Leftarrow$ perform modified Dijkstra's algorithm
  if $dist_D < 0$ or $dist_D > maxDist$ or $level_D < minLevel$ then
    $dist_D, level_D \Leftarrow$ retrieve metrics from local cache
  end if
  return $dist_D, level_D$
```

Fig. 4. Compute distance locally

```
Require: $dist_{res}, level_{res} \leftarrow performLocalDijkstra(gatheredDigraph)$
Require: $dist_{res} \in< 0, +\infty >$ and $level_{res} \in< 0, 1 >$
  $path_{new} \leftarrow$ sequence of foaf:knows triples
  for all $nodes \in path_{new}$ do
    notify user's registration server that user is cached
  end for
  $localCache \leftarrow path_{new}$
```

Fig. 5. Creating local cache of social network

4.4 Creating and Maintaining Local Cache of Social Network

The third step of user rights' computing can result in a huge digraph and expensive overload of the network. To perform the third step as rarely as possible a caching algorithm must be introduced (see Def. **??**). The goal is to remember the result of the complex distance computing. Remembering all information gathered from other services would provide a lot of redundancy and could result in data inconsistency. The local cache keeps only paths between two nodes in the digraph D_{SN} which could be used in the first or the second step of distributed user rights computing.

Definition 6 (Local Cache of Social Network). *Each UPM_{SN} maintains a local cache of social network $LCSN$ consisting of some edges $r \in R_{SN}$ between vertexes $m(r)_A \in M_{SN}$ and $m(r)_B \in M_{SN}$ in the digraph model of social network D_{SN}.*

System creates a cache (see Fig. **??**) by adding new paths to local store. Registration servers of all users that were represented by outgoing vertexes in the added path, are notified about the caching procedure. If some friendship information about the user has been changed, RS_{SN} sends update notification to services that maintain the cached information. The service that receives this notification invalidates cached path starting from the node representing the user on whom the information has been changed.

5 D-FOAF - a Distributed Identity Manag on Social Networks

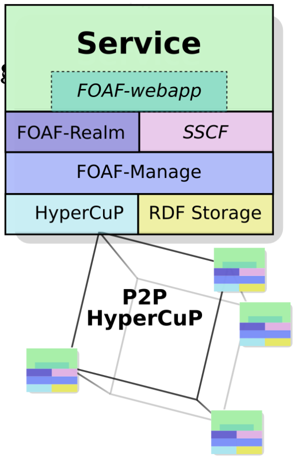

Fig. 6. Architecture of the D-FOAF system

The concept of a distributed identity management system has been implemented in the FOAFRealm project [?,?]. FOAFRealm delivers a plug-in for Tomcat [?] JSP container and utilises FOAF [?] metadata extended with concepts required by distributed user profile management on social networks. The main feature of FOAFRealm is the implementation of `org.apache.catalina.Realm` and `org.apache.catalina.Valve` interfaces that introduce the concept of *Community Driven Access Control* (see Def. **??**) and *Distributed Community Driven Identity Management* (see Def. **??**) to J2EE web applications. The use of FOAFRealm core features like authorisation and access rights management is transparent to the web application builder. FOAFRealm encodes access control definitions in a form of literals that are understood by Tomcat as realm group definitions but are processable by FOAFRealm. Example **??** shows how the *Social Networked Access Control List* (see Def. **??**) is encoded in FOAFRealm.

Example 1. ACL restricting access to a resource to the network of people that are within `3` degrees of separation from the user `sebastian.kruk@deri.org` and whose trust level computed across the social network is above 50%, can be encoded in FOAFRealm as `F[mailto:sebastian.kruk@deri.org]3,5`, where 3 stands for 3 degrees of separation and ,5 represents the 50% minimal trust level.

5.1 Architecture

D-FOAF, Distributed FOAFRealm, utilises the HyperCuP P2P infrastructure to connect and exchange information between FOAFRealm instances. There are four major features supported by D-FOAF:

- Distributed user authentication (see section **??**)
- Distributed identity management (see section **??**)
- Secure distributed computing of distance and friendship level between users (see section **??**).
- Social semantic collaborative filtering [?]

The current implementation of FOAFRealm consists of four layers (see Fig. **??**):

- The distributed communication layer provides access to a highly scalable HyperCuP [?] Lightweight Implementation [?] of a P2P infrastructure to communicate and share the information with other FOAFRealm implementations.

– FOAF and collaborative filtering ontology management. It wraps the actual RDF storage, providing simple access to the semantic information from the upper layers. The Dijkstra algorithm for calculating distance and friendship quantisation is implemented in that layer.

– Implementation of the `Realm` and `Valve` interfaces to easily plug-in the FOAFRealm into the Tomcat-based web applications. It provides authentication features including autologin based on cookies.

– A set of Java classes, tagfiles and JSP files plus a list of guidelines that can be used while developing a user interface in personal web applications. This layer includes general user interface implementations for user profile management, social semantic collaborative filtering and multifaceted browsing.

5.2 User authentication in D-FOAF

To provide a single registration feature in the whole federation of FOAFRealm services (see Def. **??**), D-FOAF performs a distributed authentication algorithm (see Fig. **??**). When a user logs in for the first time, the service locates his/her registration server by sending a registration server discovery broadcast query over the HyperCup P2P network. Once the location of the registration server is found a local user profile is extended with the triple <`user_mbox`> <`foaf:seeAlso`> <`registration_service_uri`> indicating the location of the registration server to speed up authentication operations in the future. Authentication responsibility is later delegated to the user's registration server, which answers with the user's profile upon successful registration, or indicates that the supplied credentials are wrong.

5.3 Distance and Friendship Level Metrics Computing in D-FOAF

Computing distance and friendship level over a distributed RDF is required for evaluating user access rights, and is probably one of the most complex algorithms in D-FOAF. The system has to cope with a variety of problems. The problem gets less trivial when the FOAF graph is distributed among many services consisting the D-FOAF network. The distances computation is performed in three steps implementing the algorithm defined in section **??**:

Step 1 A single instance of FOAFRealm implements the modified Dijkstra algorithm to compute the distance and the friendship level between users. Computations are performed on the local FOAF database.

Step 2 The distance and friendship level computation algorithm is performed on each node of the D-FOAF network independently. The query is send as a broadcast on the HyperCuP P2P backbone of the D-FOAF network.

Step 3 The system has to gather all the information, required to compute the distance into one place - the FOAFRealm instance that invoked the query. The complete information about the profile of the first user is retrieved. Next all <`foaf:knows`> triples describing direct friends of this person are gathered with the HyperCuP broadcast. Local server builds temporary FOAF database and performs standard local computation together with retrieving missing <foaf:knows> profile information on demand.

Caching The third step might generate a huge RDF graph and expensive overload of the network with broadcast messages. The caching feature has been implemented in the D-FOAF to address these issues. Since the original social network in each FOAFRealm node is signed by the registration servers, <`foaf:knows`> triples that builds up the cached path are stored in a separate RDF store not to weaken the previously introduced security mechanism.

5.4 Evaluation

We evaluate FOAFRealm against current distributed identity management systems. Firstly, Microsoft Passport[?] gives a simple Single Sign-On feature. Because of the centralised topology, proprietary status and very frequent bug reports, the system has not been yet widely accepted. Moreover, users do not want to share their private information to a commercial company. The solution cannot guarantee that the privacy information will not be used for illegal purpose.

To solve this privacy problem, the Liberty Alliance Project[?] suggests open specification and multiple identity providers. The more than 150 organizations are bringing together their specification. They have also added ACL features based on social relationships. The project is targeted at larger scale and more business oriented web services and thus it is used very rarely in small enterprises. Users find it hard to make their own server and they still need to relay on large organizations. Moreover, the specification needs complicated procedure to make social relationships without opening the personal information of other people.

The SXIP[?] makes a more simple solution to support privacy. It is a lightweight and open source solution. It also gives a development kit so users can make their own private servers to save their private information. Despite the fact that it is a step forward with respect to Passport, Sxip is still centralised from the perspective of the home server. However, SXIP does not give any access control list or social relation features.

We have described the identity management and social network features of FOAFRealm. It is also an open source solution and users can have their own FOAFRealm servers. However, at the current stage, FOAFRealm exports their relationship information to other FOAFRealm servers, which can be private information and future work is needed to research how to prevent abuse of sensitive data.

To futher test research presented in the previous sections we have deployed FOAFRealm in JeromeDL [?,?]. DERI has decided to use JeromeDL (and FOAFRealm) as their main digital library engine. Websites http://library.deri.ie and http://library.deri.at successfully serve digital publications, offering distributed FOAFRealm profiles management, fine grained access control lists and semantic searching. FOAFRealm features showed to be useful for majority of users and most of them quickly adopted sharing bookmarks with friends.

6 Related Work

Our work not only introduces an interesting approach to common problems but also integrates several existing concepts. Two most fundamental are user management and social networks research areas.

6.1 User Management

Project Integration Architecture [?] researched by NASA, provides a distributed user management and access control lists. Problems of the security were considered and described for the whole process of authentication [?]. The solution, which was implemented in CORBA [?], is based on distributed lock management [?] and deadlock detection. Unfortunately, the system does not support any semantic user profile description like FOAF.

The EMBASSI [?] project propose an original approach to distributed user profile management which uses agent based architecture. The system divides user profile into two types - the personal generic user data and domain values that are relevant for specific environment. It has been shown that this approach leads to a compound set of generic user variables and it can meet the requirements for different application areas.

Identity 2.0 [?] is a protocol for exchange of digital identity information. The general idea is to provide users with more control over what others know about them. The next version of the system mentioned above - Sxip [?] will provide increased anonymity for users. Furthermore, it will be possible to adjust security needs to specific site.

MyProxy Credential Management Service[?] initiative has already solved the problem of managing different user accounts. But the work was conducted in the context of Grid and the users are not enabled to take advantage of existing social networks and semantic user profile description.

The SD3[?] is a distributed trust management system that introduces high-level policy language. The system utilising groups and permissions instead of access control lists and social networks and that is the main difference between this project and D-FOAF.

An interesting approach was proposed in PeerTrust Project[?], which concerns a decentralised Peer-to-Peer electronic community. The important contribution of these authors is to build a trust model based on only three factors: the amount of satisfaction established during peer interaction, the number of iterations between peers and a balance factor for trust. And the trust model is the main difference in comparison with FOAFRealm system.

The idea of distributed user profile management become more and more popular it results in projects developed by open source community. Drupal [?] offers distributed authentication module and Single Sign-On feature. XUP [?] takes advantage of XML format which holds user account information. This issue competes with the W3C FOAF [?] metadata recommendation.

6.2 Social Networks

The six degrees of separation [?,?] theory began the research and development of social networks. The number six derives from an experiment performed in 1967 by social psychologist Stanley Milgram [?].

Because the Milgram's experiment had been rather small, it was questioned. As a result some sociologists [?] recruited over 60,000 participants from 166 different countries and they performed tests on the Internet environment.

The first website called *HotLinks* which utilised the concept of the six degrees of separation was published in 1998, and was available for four years. Then, the members were moved to Friendster [?] network, which was founded in 2002. Since winter 2002 Friendster network is becoming more and more popular. There are more than 21 million members at the moment.

Nowadays, there are a few dozen networks that take advantage of six degrees phenomena. They differ in many ways. For example, Hungarian WIW [?] and Orkut [?] projects require an invitation in order to join the network, which guarantees that at least one relationship with community for new members, while it is not necessary in Friendster mentioned above. In addition, we noticed recently a large grow of business oriented networks, like e.g. LinkedIn [?] and Ryze [?], that manage professional contacts, enabling users to find employer or employee.

Complexity[?] is an on-Line journal. An special issue published in August 2002 was dedicated to the role of networks and network dynamic. Although, the focus was on showing complexity for different levels of network architecture, a large part of the journal was related to social networks. The mentioned issues were helpful in comprehension of network-based analyses and explanations.

The scope of social networks is much wider. Recently, the idea was adopted in order to protect from spam, which becomes such a ubiquitous problem. Introducing reputation networks and taking advantage of Semantic Web, TrustMail project [?] extends the standard social network approach. Moreover, various algorithms were considered and a prototype email client was created and tested. It resulted in highly accurate metrics. Additionally, valid e-mails from unknown users can be received, because of connection in the social network.

7 Conclusions and Future work

We introduced the identity management based on social networks. We showed how utilising of the social networks in the identity management systems can reflect in high granularity and scalability of the access control features providing notion of access rights delegation. We detailed algorithms for the distributed identity management that have been implemented in the FOAFRealm/D-FOAF project presented in this article.

Although the FOAFRealm system presents a complete solution for distributed identity management based on social networks, there is number of issues that are being implemented at the moment. The access rights delegation based on the social network information and trust levels has been so far tackled within a single context. Further research on multiple contexts of trust levels and distributed trust computation will be carried on. The idea of single identity registration can only be realised when a lot of online services can use or connect to the D-FOAF network. To make that possible, implementations for other platforms like .NET or PHP will be provided in the future. In addition the third step of evolution of FOAFRealm system, called DigiMe, has been initiated. The goal of DigiMe project is to deliver a complete solution for mobile devices. This solution will not only provide access to existing D-FOAF networks but provide users with better control over their profile information. DigiMe will enable users to store this information on the mobile device. Finally, further research on algorithms for distributed FOAF computations including security, caching and replications will be continued.