HelloWorld: An Open Source, Distributed and Secure Social Network

Markus Ackermann
maac0001@stud.fh-kl.de

Krister Hymon
krhy0001@stud.fh-kl.de

Benjamin Ludwig
belu0001@stud.fh-kl.de

Kai Wilhelm
kawi0002@stud.fh-kl.de

University of Applied Sciences Kaiserslautern
Amerikastraße1, 66482 Zweibrücken, Germany

Abstract

HelloWorld is an open source project that aims at the development of a distributed and secure social network. The difference to existing social network platforms, like MySpace or FaceBook, can be summarized in three main features.

Authentication: Instead of having to create an ID bound only to HelloWorld, the user will be able to use an independent and free authentication system.

Decentralization: HelloWorld will be able to exist without a central data base. The user will be able to decide on which server his data will be located or even distribute it to different servers.

And security: With respect to the human right of informational self-determination, all the data will be protected by most modern cryptography technologies.

HelloWorld will be the first true social network that deserves to be called social, because all the data will be controlled by the users and not by the providers of the network.

1. Introduction

First we will have to make clear some fundamental concepts and definitions concerning digital social networks in the internet.

1.1 Definition of a social network

Here *Wikipedia* gives a very good definition of what a social network is:

„A social network is a social structure made of nodes (which are generally individuals or organizations) that are tied by one or more specific types of interdependency, such as values, visions, ideas, financial exchange, friends, kinship, dislike, conflict, trade, web links, sexual relations, disease transmission, or airline routes. The resulting structures are often very complex.“[1]

Let‘s think about that definition. There are two fundamental elements a social network consists of. Nodes which are represented by people in our case and connections between them which we will call relationships.

For example a circle of friends is a social network consisting of people who are connected by a friendship relation. Further examples of social networks are business colleagues or families. The last two examples show that people can belong to social networks without having a choice.

The ways to join or leave a social network highly depend on its structure. Joining a soccer club for example is just a formality if the team coach accepts you, but joining a family can only happen through birth or marriage.

Important is also the fact that everyone belongs to several social networks sharing different personal facets with each. These facets can be everything from appearance over hobbies to the most intimate desires.

1.2 Social networks in the internet

The internet extended social networks by new ways of communication like email, chat or forums. These techniques make it easier for people to stay in contact and to maintain their relationships.

Profiling oneself changed completely. The anonymity of the internet makes it possible for users to decide which information about their personality they want to share with others, for example on their own home page or in chat rooms. Users even can create one or more fake identities and play someone else, which is an attraction for many users. In our eyes this is an misuse of the internet concerning the idea of a social network. But we‘ll talk about that aspect later.

In the end the internet makes it easier for people to communicate and to profile themselves. Therefor it is a great advance to social networks.

1.3 Summarizing the core functionality

In the following we will look at social networks from the point of view of individual persons. Because that is exactly the way people experiences social networks around them.

What are the most important actions people perform concerning a social network in the internet?

People in the same social network communicate with each other and share information about themselves. The relationships between them are defined by the very information they share to each other especially personal information. For example: if you send an email to a business partner who is not a member of your circle of friends or your family you probably wouldn‘t attach your private phone-number. Instead you would attach your contact information of your bureau.

People can extend their social networks by sharing the same information they share with other members to new ones.

1.4 Important key words

Before we go on, we have to make clear some important key words.

Profile: A profile is a collection of personal information defined by the user which forms his identity in the internet. As mentioned before a user can create fake identities. But we now will assume that each user has only one identity which consists of true information.

Sub profile: Because a user doesn‘t want to share all his information to all the users he is connected to, he is able to create one or more sub profiles. A sub profile is a subset of information taken from the profile. According to the example with the business partner above, creating sub profiles is a very natural decision, people make when interacting with others. They filter out information from their profile according to the type of the relationship.

Relationship: A relationship describes a connection between two users. From the point of view of one user, the relationship is defined by the sub profile(s) which he shares to the other one. Relationships are not transitive. That means, if A is related to B, and B is related to C, but A isn‘t related to C automatically.

2. Current situation

A few years ago, a new form of social networks appeared in the internet, which are called social networks or services.

These services bring together all the fundamental functionality that is needed for users to maintain their relationships and their profiles and to communicate with each other. All these services are very similar in use and appearance. These are best demonstrated by looking at one example. For this we chose *Facebook*, because in the meantime it became one of the biggest social network platforms.

2.1 Example: Facebook

Facebook is mainly addressed to students. They can create an user account with their email-address as unique identifier. After that they can access the platform with their email-address and password and a chosen nickname. The platform offers different forms to create a profile. This forms are limited to special types of personal information like contact information, hobbies, photos etc.

At the beginning it is not possible to see the profile of any other users. From our point of view, joining the network platform is therefor not equal to joining a big social network. It is only possible to see the profile of other users, if two users establish a relationship to each other. In *Facebook* relationships are called friendships. Two related users are called friends.

A friendship can be established, if one user invites another user to be his friend, and the other user accepts that. A search function allows users to search other users by their nickname. Friendships can be canceled by one participating user.

After that it is possible to see the personal information of each other. It is not possible to determine which profile information are visible to friends. That means every friend of a user sees the same information. Friendships can be transitive, if a user allows the friends of his friends to see his profile. This transitivity is not necessarily bidirectional.

Facebook offers different ways to communicate to each other. Users can send messages to other users, which are only visible to them. Or they can write in the guest book of a friend. Those messages are then visible to every friend of that special friend.

Facebook has by now over 60 million active users. To manage all that data, *Facebook* uses one or more servers to store it. To the providers of *Facebook*, all the users data and all their friendship connections are visible all the time.

2.2 Some facts and assumptions about usage

2.2.1 The users

Here we will take a look at another example: the german *StudiVZ*. This is also a social network platform addressed to students.

In difference to *Facebook* the users profiles are visible to every other user by default. The visibility can be limited, but most users do not realize that. In fact most users do not really care about spreading their personal information. A statistic from december 2006 shows, that at this time, there were 1.074.574 profiles hosted, but only round 40.000 weren‘t visible to everyone. [2] These are only round 4%. 708.000 were defined as active profiles, that means they have at least two friends, have a visible profile and are at least part of one group. It is not difficult to create a fake profile, because you just need a valid email-address. So the number of an active profile probably contains without much doubt many fake profiles. A fake profile means that no true information about the user are contained in the profile.

The question is: Why should someone create a fake profile? For example to crawl all the public visible data without leaving a trace to the crawlers identity. In this case crawling means to collect all the available user profiles and store it. And this has definitely happened in the past! For example some students from our university did it and the statistic we used before are also from someone who crawled *StudiVZ*.

Another aspect about the usage of social network platforms is that many users are active on more than one platform. Transferring their profiles or relationships from one platform to another is hardly possible. On every platform the user has to create a new profile and has to establish all relationships again although they are already existing.

2.2.2 The providers

Now we will look at the providers of social network platforms. Once again, we take *StudiVZ* as an example. At the beginning of *StudiVZ* the general terms and conditions of *StudiVZ* forbid the usage of user data to the providers. In december 2007, *StudiVZ* changed them allowing the providers to sell the users data for personalized advertising. The users were informed about that and had to accept that. Otherwise the account would have been deleted. At this point it became clear, where the business model of the providers aimed at. To collect a great amount of users and then sell the data to the industry. In fact *StudiVZ* was developed by three students and was sold in january 2007 to the german *Holtzbrinck-Konzern* for around 80 million euros. The connection between the sale and the change of the terms of the conditions is obvious.

That shows that for the providers of social networks platforms there is a lot of money to make with the users personal information.

Each user means pure capital to the providers.

2.3 Problems

The first problem is that users are not able to define sub profiles. They only can decide who sees their profile, not which parts of it.

Another problem is, that a user cannot transport the data from one network to another. Connecting this and the fact that every user is pure capital to the providers of social network platforms make clear, that providers just don‘t want the users to be active on other social network platform. Or at least make it difficult for them. Furthermore there is no way to identify a user across several platforms.

The fourth problem is that transitive friendship relations (for example at *Facebook*) make it very difficult for the users to control who has access to their profiles. *StudiVZ* has an even worse problem, because most users have public profiles which are visible to everyone. As we have shown below, crawling the data is not a problem at all, so there is a big lack of security. And if crawling wouldn‘t have been possible, guess who has still access to the data and the relationships?

Apart from the terms and conditions there is no way to control what the providers of the social network platforms use the data for. The centralization of the personal data of millions of users and their relationships offer to the providers an unbelievable amount of personal information. Even more than any security agency or other government organization ever had about the citizens. We talk about contact information, hobbies, interests, religious believes, political attitude...

From the moment on an user creates a profile on a social network, he loses the completely control over his data. Assuming that someone has a profile on any social network platform. Can he be sure that for example an employer he applies to is not able to check his personal information? And what if those information lead to the users rejection?

The next problem comes with the capital social network platform providers make with the data of their users. In fact they use it for personalized advertising and make a lot of money with it. But the users don‘t get anything back for it. Instead they pay for the service with their personal information. Of course they can use the many functionalities of the social platforms, but the internet already offered enough ways to communicate with each other and maintain social networks.

At least there is another problem concerning the centralization. Users of a social network platform have to a access a web page to use the service. Communication between the client (browser) and the server (social network platform) often is not secured for example with HTTPS.

To sum it all up, there is a big need of functionalities that social platforms offer. But the existing ones deal with immense problems concerning identification, security and data centralization.

Users have no chance to prove their identity across several platforms. They can not be sure of the security of the data. They cannot control what the data is used for and they cannot decide where the data is stored.

2.4 Solution approach

By looking at the current situation of social network platforms and the problems they deal with, it is very easy to create an abstract solution approach:

The solution is to develop a free, secure and decentralized social network connected to an open and independent identification system. Including all the fundamental functionalities existing social network platforms offer.

And this is exactly what HelloWorld is going to be. HelloWorld will be a client software which runs on the user‘s local machine. It is simple impossible to provide the security technologies which we will describe later as a web service.

And the best thing is that all the technologies to create it already exists and only have to be combined.

3. Authentication and identification

Let‘s talk about the problem of identifying users in the internet. This problem exists since the internet was born (remember the fake profiles). A better term than identification in the technical case is authentication.

3.1 Problem

In most cases when you register in several web services you have to sign in with your email-address, because the email-address is always unique. But a throw-away-email is easily created and is not an identity for someone like a password id in the real world.

3.2 Technical solution

With the concept of *OpenID* the idea of a distributed digital identity management was born. *OpenID* is a decentralized system which offers an authentication protocol that is based on well established open web standards (URI, HTTP, SSL, Diffie-Hellman).

But there are some problems in this concept, as you need an OpenID-Provider, a server that hosts your identity data and realizes the authentication protocol. The main problem is, that your identity data is stored unencrypted on this server, and so you give the control of this data to someone else. Sure, you can be your own OpenID-Provider but how many users own a server to realize this?

Furthermore the OpenID-Provider is able to collect very sensible data of your web-activity, as he is engaged in every authentication process that uses *OpenID*.

What we need is a decentralized protocol to authenticate users without the need to give sensible data to someone else. Based on the concept of *OpenID* we developed our own authentication protocol called HelloWorldID. Equal to *OpenID* we will use a URI as unique identifier, e.g. *username*.helloworld-network.org or helloworld-network.org/*username.* This identifier can be hosted on every domain and so it is not bounded to a special server. This URI will point to a HTML page, that contains a meta link to an XML-representation of the owners RSA-public key. This public key will offer web-services the ability to prove the authenticity by themselves, so there is no need of a third party to be involved, as it is in *OpenID*.

The protocol will work like this:

1. The user gives the web-service he wants to login his HelloWorldID URI via a HTTP-Request.

2. The web-service downloads the XML-representation of the users public key via the meta-link and parses it into the corresponding runtime object depending on the programming-language he is using.

3. Then the web-service generates a random phrase and encrypts it with the users public key, the unencrypted phrase will be stored in a session.

4. The encrypted phrase will be sent back to the user via a HTTP-Response.

5. The user decrypts the phrase with his private key and generates a signature as well.

6. Then he sends the unencrypted phrase and the signature back to the web-service.

7. In the last step, the web-service checks the phrase and the signature and is able to authenticate the claimed Identity by the result.

As you can see, the protocol has no need of any sensible data and doesn't even need a password or something like that, and, as mentioned above, no third party is involved in this process. It only uses the secure concept of asymmetric encryption.

4. Decentralization

Decentralization means, that you have several storage mediums instead of one. In this case we are talking about internet server or provider.

Remember the three problems which occurred by a central storage of all the users information by social network platforms.

4.1 Technical solution

The sub profiles are available by links, which have to be shared under the users who are starting to establish relationships between each other. Users are able to choose on which server they want to upload their sub profile(s). This could be for example their own web space, the web space of a friend or any other provider in the internet. Of course there will be a stable network of HelloWorld servers which make it possible for the users to upload their files on it if they don‘t have an own web space.

A legitimate question at this point is, where is the real difference to already existing social network platforms? Most social network platforms earn money with the personal information the users upload to their own server or provider.

Well, the main difference between a sub profile of HelloWorld and profiles of other social network platforms is, that the HelloWorld sub profiles are encrypted. This is explained in the next topic under security.

But with the solution of decentralization, one problem is coming up. The most social network platforms have one big database where all the users are stored. So it is easy for them to offer a search function to find other users. But how can you find other users in the internet while all the users are distributed all over the internet? This question is answered under the phone book question in chapter 6.4.

5. Security

As mentioned above, the user should have the full control over all his data and he should be able to decide with which users he wants to share his data with. That means that his account, his sub profile(s) and all the communication with other users have to be secure.

For this reasons, HelloWorld will use different encryption methods.

5.1 Problems

There are three questions: How can the user‘s account be protected? How can the uploaded sub profile(s) be protected from being read by unauthorized persons? And finally, how is it possible to guarantee a secure communication channel between users?

5.2 Technical solution

When using HelloWorld for the first time the user has to generate an account on his local machine. This account consists of the user‘s profile, his contacts and his personal configuration. It is encrypted with a password and two values (a salt and an iteration count) which are generated automatically by the system. This is necessary to guarantee a better protection of the account, for example against brute-force attacks.

Every form of communication between the users is secured by an combination of asymmetric and symmetric encryption. Every user has a key pair which includes an public key and a private key. The private key is only known by the user himself. The public key is accessible to every user of HelloWorld and is the first step to enter a secure communication. He is like an instruction for a treasure chest, where every user can hide something in it (a message etc.). Only the owner of the private key which belongs to the chosen public key is able to open it and he can take out the content.

The detriment of the asymmetric encryption is, that it is very CPU-intensive. So it is usually used it to exchange a key for a symmetric encryption.

The symmetric encryption uses the same key for de- and encryption. So if two users start a communication with each other, they exchange a symmetric key by using the asymmetric encryption. The following messages or data is than encrypted with the symmetric key, which is generated for every session. In HelloWorld this key is called the session key.

The encryption of the user‘s sub profile(s) works in a similar way. For every sub profile, an own symmetric key will be generated. This key is called the profile key. The user saves this key, and can share it with any users he wants to share his data with. Neither the server provider where the user hosts his sub profile nor other users who are not allowed to see the user's sub profile will be able to gain access to his data.

HelloWorld will also use digital signatures for any messages which are send between users. A signature is generated over the hash code value of the message content. The sender uses his private key to encrypt the hash code value and the receiver decrypts it with the sender‘s public key. After that, the receiver himself generates the hash code value over the message content and matches it with the decrypted once of the sender. Only if they are similar he can be sure, that no one changed the message before. This procedure also happens by the exchange of session or profile keys.

6. Communication

To offer users the possibility to communicate to each other in a secure way, a simple protocol will help to exchange messages. There will be different types of messages and two ways to exchange them. Let‘s first take a look at the protocol.

6.1 Protocol definition

First to mention here is the important fact that the protocol highly depends on the availability of the users public keys. This will be realized with the HelloWorldID concept described in chapter 3.2

The piece of data that people will exchange is called message wrapper. This wrapper always consists of an encrypted message, an encrypted session key, and signatures for both to prove their authenticity and integrity.

The session key is encrypted with the receiver‘s public key, so only he can decrypt it with his private key. Then this session key can be used to decrypt the message. The message consists of a head and a body. The head gives information about the sender, the location of the sender‘s public key and the type of the message. The message type defines how the content of the body has to be interpreted. For the beginning there will be 3 message types:

Text message: A simple text message

Relationship request: A relationship request contains the sender‘s sub profile location and the profile key to read it. The sender offers his personal data with the request. Without a central instance it s not possible to control whether both users agree to the relationship. This is like relationships work in reality. If you give your private phone number to someone else.

Relationship response: A relationship response is like the relationship request, but makes it possible to inform the user that this message is the response to a specific request.

After a successful exchange of a relationship request and response the users can be called related to each other. The type of the relation defines whether this relation can be called friendship, business or whatever.

The communication channels used to exchange the message wrappers will be email and peer-to-peer. Of course an user can also burn a message wrapper on a CD and send it to the user he has created it for by mail. But that functionality can hardly be integrated in our software. So let‘s take a look at email and peer-to-peer.

6.2 Email

Of course a user needs an email-address to use this functionality. He can enter his mailbox account data and save it to his HelloWorld account. After that he can send and receive HelloWorld message wrappers directly with the integrated mail client. The application adds an identifier to the subject field of each email that makes it possible for the receiver‘s application to check his mail box for HelloWorld messages.

Those messages are then downloaded and deleted from the mail server. They are stored encrypted in the users account on his local machine. That makes it possible to view a history of received messages and react on them at any time.

6.3 Peer-to-peer

An integrated peer-to-peer client will offer the possibility to send and receive messages directly. Text messages for example can then be decrypted at the time they arrive and directly shown to the user. This way we can provide instant messaging and data exchange in a secure way.

Finding other peers without a central data base is not easy at all. But there is already an open source approach that deals with that problem. It is called *Freenet.* Here is a short description taken from the projects home page:

„*Freenet is free software which lets you publish and obtain information on the Internet without fear of censorship. To achieve this freedom, the network is entirely decentralized and publishers and consumers of information are anonymous. Without anonymity there can never be true freedom of speech, and without decentralization the network will be vulnerable to attack.*

Communications by Freenet nodes are encrypted and are "routed-through" other nodes to make it extremely difficult to determine who is requesting the information and what its content is“ [3].

We will integrate this technology in our software to benefit from its great advantages.

6.4 The phone book question

HelloWorld does not have a central database where all the users information are saved. So it's impossible to offer a list of all HelloWorld users and make it possible to find each other if you didn‘t know each other yet.

The idea for this problem is very simple. If you create an account at HelloWorld you are able to sign in to the telephone book and show some basic information about you. Basic information are something like your user (nick-)name or some contact information like email-address etc., which makes it possible to other user starting a communication.

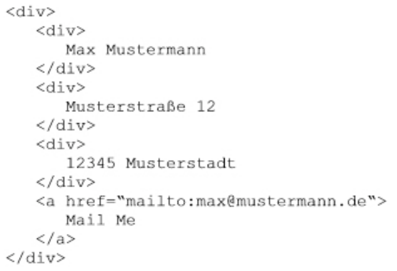
Another add-on could be the creation of special network-groups, where users are able to sign in their sub profiles. This could offer a global search for users which are sharing the same interests or hobbies. This service requires a network of decentralized HelloWorld servers.

7. Profiling

In HelloWorld profiling will become more easy and extendable. We will provide interfaces to open standards so that users can import and export profile data. One of those open standards are microformats like *hCard*, which is the HTML-representation of *vCard*. These formats we will described in the following.

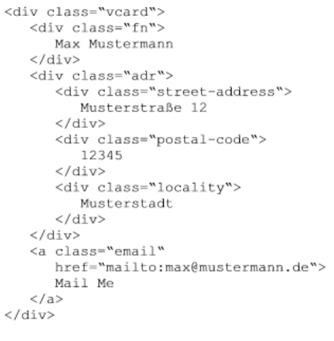
7.1 Microformats

Microformats are simple additional information which provides the semantic meaning of data. Defining a *vCard* in HTML can be generated like this:



```
<div>
   <div>
      Max Mustermann
   </div>
   <div>
      Musterstraße 12
   </div>
   <div>
      12345 Musterstadt
   </div>
   <a href="mailto:max@mustermann.de">
      Mail Me
   </a>
</div>
```

By using microformats, a user can place semantic meanings into his *vCard*. Now machines can understand what the user has placed into his web page. It looks like this:



```
<div class="vcard">
   <div class="fn">
      Max Mustermann
   </div>
   <div class="adr">
      <div class="street-address">
         Musterstraße 12
      </div>
      <div class="postal-code">
         12345
      </div>
      <div class="locality">
         Musterstadt
      </div>
   </div>
   <a class="email"
      href="mailto:max@mustermann.de">
      Mail Me
   </a>
</div>
```

These simple conventions make it possible to realize imports and exports of existing business cards or other standardized data.

8. Bringing it all together

So what we are developing will be a solution, that offers the ability to combine almost every kind of digital communication, that are in widespread use today, in one piece of software. That means HelloWorld will combine the features of social networks like *Facebook*, instant messengers like *ICQ*, voice-chat and video-chat like *Skype* and file sharing in one application and protocol.

Furthermore all communications will be secure and the user will have the responsibility to build his own chain of trust by key distribution.

8.1 Example use case

Lets take for example Bob. Bob is a student of computer sciences and is a registered user in *Facebook* and *Orkut*. He also uses *ICQ* and *Skype* to talk to his contacts. Many of his contacts are also users in all of these platforms. So why must Bob use two different webservices and two different desktop applications to cover his social activity if he could be able to manage all in one solution? And why should he give his personal data to four different providers? Maybe he manages different kinds of relationships in the different platforms, e.g. in *Facebook* his student colleagues, in *Orkut* his business partners and in *Skype* and *ICQ* his nearest friends.

So now Bob decides to change to HelloWorld. The first step after the installation is to manage his own identity, that means define the set of his personal attributes he wants to share. After that he defines what kind of relationship types he wants to establish, and for every relationship type he defines the subsets of his identity visible for those relations. In Bobs case these are three types: student colleagues, business partners and friends. For the business partners e.g. he decides to share his skills, work experiences and basic personal information like his age and his location. His friends will be able to see much more like his political attitude, free-time activities, his favorite movies and what kind of music he listens to. For his study colleagues he publishes e.g. what lectures he attends to. Then he loads these sub profiles encrypted on his own server.

Furthermore he creates his own HelloWorldID as a sub-domain on his server. Now he tells his contacts to do the same and they exchange their HelloWorldID‘s over the other platforms. Now they can reestablish their existing social graphs in the HelloWorld network.

After that, Bob only needs one application, and he must not fear what will happen with his data.

9. Current project status

Currently the project can be found at *Sourceforge* with the URL http://sourceforge.net/projects/ opendude/. The development has already started. A prototype has been developed to test parts of the functionality.

9.1 Prototype

The prototype we have developed already offers some important functionalities. The user can create an user account on his local machine and manage his profile. He can import a *vCard* and define several relationship types and the corresponding sub profiles.

All the security techniques we described are already integrated and working. FTP, SFTP and a mail client are integrated too. So it is possible to upload a sub profile and send a friendship request to another user. Of course the exchange of the public keys is still a problem to be solved.

10. The future of HelloWorld

The next steps are clear. We are going to work on the phone book question and will integrate a peer-to-peer client based on *Freenet*. The prototype of the user interface will be advanced. More functionality like guest books and photo galleries will be added.

To not make it unnecessary for the user to run an extra software on his machine, we will develop several plugins for highly distributed products like *Mozilla Thunderbird* and *Firefox*.

Everyone who is interested in the development of HelloWorld is encouraged to take part in this project. *SourceForge* made it possible that there are already some collaborations running with developers around the world.

The final goal is to provide a user friendly piece of software that is totally free and has the power to replace all the existing social network platforms by providing a secure solution to all their problems. HelloWorld is going to be the first true social network that respects the user‘s right of informational self-determination.

11. References

[1] Wikipedia contributors, "Social network", Wikipedia, The Free Encyclopedia, http://en.wikipedia.org/w/index.php?title=Social_network&oldid=222956055 (accessed July 1, 2008)

[2] Hagen Fritsch, "StudiVZ - Inoffizielle Statistiken vom Dezember 2006", http://studivz.irgendwo.org/, 9. december 2006, (accessed July 3, 2006)

[3] Ian Clarke, what is Freenet?, http://freenetproject.org/ (accessed July 3, 2008)