Survey on Online Social Networking Security

Arpitha P M*, Janhavi V
Dept. of CSE, VVCE, Mysuru.

DOI: https://doi.org/10.21467/proceedings.1.64
* Corresponding author email: arpithamys0@gmail.com

Abstract

The recent increase in popularity of online social network applications raises serious concerns about the security and privacy of their users. Beyond usual vulnerabilities that threaten any distributed application over Internet, online social networks raise specific privacy concerns due to their inherent handling of personal data. In this paper we point to the centralized architecture of existing online social networks as the key privacy issue and suggest a solution that aims at avoiding any centralized control. Solution is about online social network based on a peer-to-peer architecture. In fully distributed nature, the peer-to-peer architecture inherently avoids centralized control by any potentially malicious service provider. In order to cope with the lack of trust and lack of cooperation that are akin to peer-to-peer systems and to assure basic privacy among the users of the social network, this solution leverages the trust relationships that are part of the social network application itself. Privacy in basic data access and exchange operations within the social network is achieved. Similarly, cooperation among peer nodes is enforced based on hop-by-hop trust relationships derived from hop-by-hop trust, from the social network.

Keywords – OSN, SNS, P2P, Client Server

1 INTRODUCTION

Online social networks are a digital representation of the subsets of relations, which the registered persons and institutions entertain in the physical world, thus forming a network graph which spans all enclosed parties and their contacts. These online social networks (OSN) are stored and maintained by usually commercial providers of social networking services (SNS) like the LinkedIn corp., xing AG, facebook, google, MySpace Inc. and the likes. The main motivation for the users to create accounts and participate in these services is sharing some information with others for some purpose. This purpose is decisive for the selection of the OSN that is used and can rather generally be classified in the two groups of [1]

- sharing professional information or
- sharing private information.

 ISBN: 978-81-936820-0-5

In the first case, the SNS usually pose as a utility to establish and entertain business contacts, and for self advertising to

www.linkedin.com

www.xing.com

www.facebook.com

www.orkut.com

www.myspace.co

possible employers and the users usually choose SNS with a rather professional touch like LinkedIn or Xing. The main purpose of the SNS in the second case is to share private information like the contact details, personal pictures, or even videos with selected groups of friends, and usually happens using more informal SNS like facebook, myspace or orkut. In both cases the OSN is democratizing the web by giving anybody the chance to publish themselves.

However, the core function with respect to the OSN is the setup and maintenance of a user's contact list, as it describes the local environment of a user's real social network and maps it into the digital graph. The contact list at the same time helps a user to keep track of his contacts (a user not only can use it as a passive register but in most of the SNS is automatically informed about any changes of his contact's statuses) and when disclosed to others, acts as a measure of the popularity of the user. These properties of SNS have led to the definition of boyd [1] according to which Social Network Sites Or Online Social Network Services are: Web based services that allow individuals to (1) construct a public or semi-public profile within a bounded system, (2) articulate a list of other users with whom they share a connection, and (3) view and traverse their list of connections and those made by others within the system. However, this definition lacks another perspective that is apparent when observing SNS: the communication of participants through message exchange, commenting on the profiles of others (or previous interactions, e.g. in recom -mendations),which merely is a message exchange with the aim to annotate the addressed profile, and the wealth of applications (starting from simple "poking" mechanisms to a variety of "gift" and "likeness" applications for interactions between users). When analyzing these SNS under the notion of their security and trust properties, a couple of threats quickly become apparent. Generally, the SNS contain a large amount of personal data on their users, which is either completely public, or partly protected to be accessible by a somewhat selected group of users only. The fact, that there are a multitude of ways to join this group for almost any user in the SNS by social engineering and in consequence to get access to even their protected data, has been shown in various studies. The fact, that more generally the protection of data in SNS is an open and pressing topic, is prevalent and has been shown in another series of studies [2],[3],[4]. Ofcourse, the user in both his ability to manage privacy controls and his awareness to the consequences of his actions (be it his privacy settings, the acceptance of a contact request, tagging users in pictures, or sending wall posts and comments to other user's profiles), presents a weak link for social networking security. However, existing

ISBN: 978-81-936820-0-5

Series: AIJR Proceedings
Proceedings DOI: 10.21467/proceedings.1

social networking services additionally contain a broad range of security issues. Possible attacks contain the pollution of the data in the social networking site, or the collection of data by third parties, which on first sight do not seem to be very dangerous. Even worse are impersonation and defamation attacks, as they may directly harm one of the involved parties.

However, taking a step back it becomes apparent, that this whole information plus a wealth of additional data on the users and their behaviour is collected and stored permanently by the provider of the SNS, which in consequence becomes a bigbrother. The importance of this fact is well represented in the virtual value of these providers, which in market capitalization ranges from 580 million US$ (the price that the news corp. payed for myspace in 2005) to 15 billion US$ (facebook's value according to its deal with Microsoft in 2007).

2 SECURITY OBJECTIVES

In the following, we pose and describe the security objectives that we require from a social networking service.

End-to-End confidentiality: Every interaction, i.e. profile lookup, contact requesting and acceptance, and any kind of message exchange has to be confidential, such that no other than the requesting and the responding party may be able to access any information on request or response, e.g. by way of eavesdropping. Proposing a decentralized system that is based on peer-to-peer techniques, messages are forwarded by a set of other peers, which may potentially contain some with malicious intent. In consequence it is important to put a special focus on man in the middle attacks, as these may be easy to mount in this environment.

Privacy: Any user needs to have the possibility to hide any personal information from the system, in the extreme case even knowledge about his participation. All information has to be hidden by default and no personal information of the user should be disclosed to any other party than the explicitly trusted contacts. Communication privacy has to be met, that is, no party other than trusted contacts may have a way to associate personal information to a network address nor maybe in the position to trace, which parties are communicating.

Access Control: Access to each SN member's account information should be managed based on the trustworthiness of parties requesting the access. Accepting or issuing a contact request may be seen as a formal operation that may disclose explicitly chosen attributes of the profile. The access control has to be as fine grained as the profile and each attribute has to be manageable separately. However, it may be possible to permit access to groups of attributes of the profile concurrently and it may be possible to permit public access to a selection of attributes to all users.

Data integrity: Both origin authentication and modification detection have to be performed for any exchanged message, be it a request or a response.

Authentication: In order to allow for confidentiality, privacy and access control it has to be possible to authenticate users and attribute messages to the users who sent them. A special

focus has to be put on the resilience to impersonation attacks, as different studies have shown that they are not only feasible, but due to the inherent trust of users into the SNS even very easy to mount. In particular, cloning attacks and forging a number of illegitimate accounts, also known as sybil attacks [8], should be prevented.

Availability: All public attributes of any profile have to be available at any time. Additionally, it has to be possible to deliver a message to any user at all times. Utilizing services in the decentralized, anonymous and partially untrusted environment, it is important to 1) enforce the cooperation of nodes, which might not have any incentive in providing a service for selected, unknown users, and 2) prevent simple denial of service attacks. Especially black hole attacks have to be avoided, as in peer-to-peer systems they may be easy to mount against the data of one or a subset of users.

3 METHODOLOGY

System Architectures of OSN: There are two paradigms of implementing an OSN in the literature, client-server architecture and peer-to-peer (P2P) architecture, which yield centralized and distributed systems, respectively.

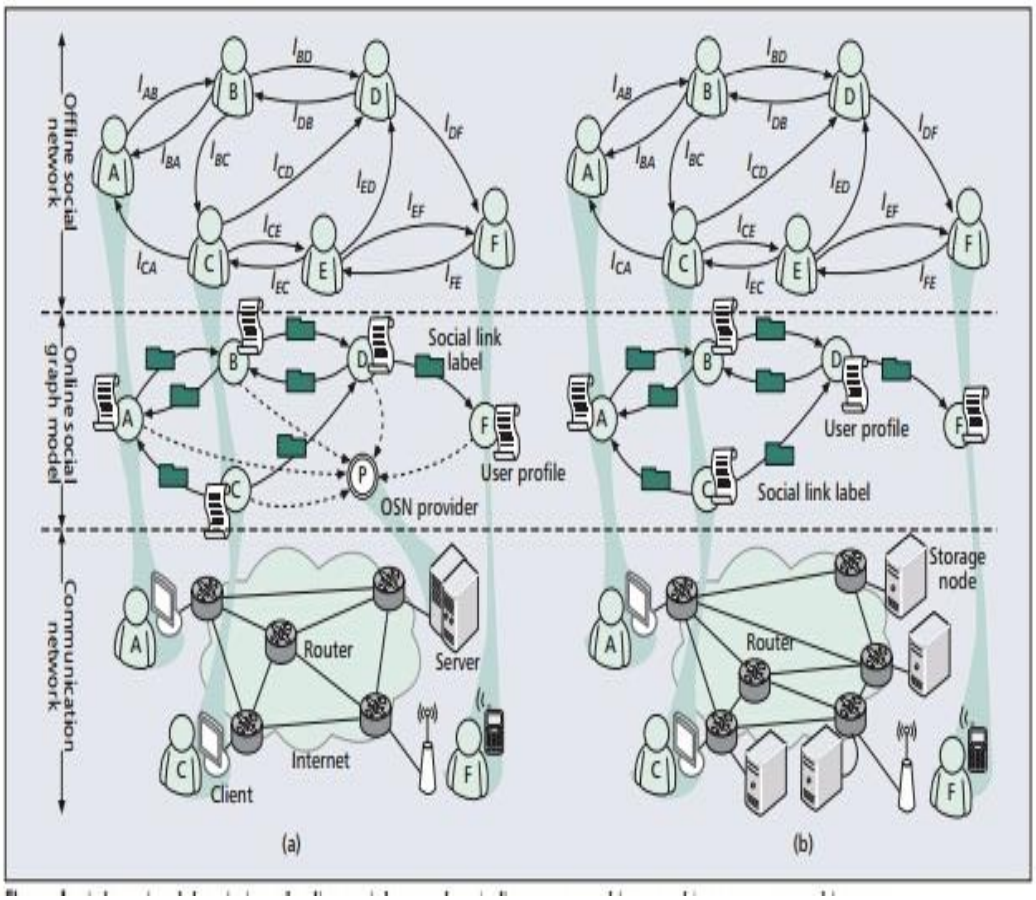

Figure 1: A three-tiered description of online social networks: a) client-server architecture; b) peer-to-peer architecture.

ISBN: 978-81-936820-0-5
Proceedings DOI: 10.21467/proceedings.1

Series: AIJR Proceedings

Client-Server Architecture: Today's OSNs are centralized and web-server based. All functionalities, like storage, maintenance, and access to OSN services are offered by the commercial OSN providers such as Facebook Inc., LinkedIn Corp., and XING AG. This traditional architecture has the advantage of being straightforward and easy to implement, while suffering from all the drawbacks of centralized systems. For example, any central entity can easily be a single point of failure, a single target for denial-of-service (DoS) attacks, and also a bottleneck for network performance.

P2P Architecture: There is a strong trend to design a P2P architecture for next generation OSNs. It adopts a decentralized architecture relying on cooperation among a number of independent parties who are also users of the OSNs. User's personal spaces are stored and maintained distributively. By supporting the direct exchange of information between devices, be it between users who have met before or between adjacent nodes of a city mesh network, a P2P architecture can take advantage of real social networks and geographic proximity to support local services when Internet access is unavailable. But for the P2P architecture, offering some functionalities (e.g., global search) of OSNs in a distributed manner is a challenging problem. Where the client-server architecture requires Internet connectivity for users to communicate via the remote central server, and the P2P counterpart supports communications via local connectivity since the role of the central server is distributed into each storage node. Note that for client-server architecture, in its corresponding online social graph model, a new entity called OSN provider is added, and each OSN user has a social (eg.trust) relationship with that provider. The peer-to-peer architecture meets the basic privacy concern through avoiding centralized control by potentially malicious application providers. Furthermore, as an underpinning of our solution, the trust relationships akin to social networks are leveraged in order to assure privacy and enforce cooperation among peer nodes. Each participant is associated with a node of this network. Each node in turn is uniquely identified by a pseudonym and a node identifier. Additionally each participant can be referenced based on its name.

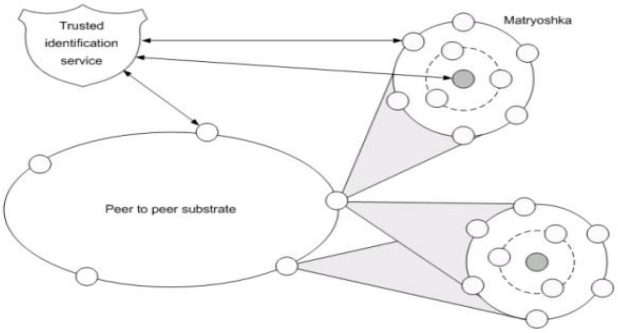

Figure 2: System main entities: peer to peer substrate, matryoshkas and trusted identification service.

System Overview :

Our system consists of three main components (see fig 2):

1) several matryoshkas
2) a peer to peer substrate (e.g. DHT)
3) a trusted identification service

Each matryoshka provides the basic distributed structure used to store the data pertaining to a user of the social network. The peer-to-peer substrate provides the global access to a user's data, based on her identifiers. The trusted identification service guarantees authentication.

1) Matryoshkas: The matryoshka of a user is a view of nodes in the system, in which nodes are located on concentric rings, centered at the core node that represents the respective user. The innermost ring consists of the set of nodes that belong to the user's trusted contacts. The second ring consists of nodes that are each a trusted contact of a node on the first ring. Further rings are built through similar trust relationships. It should be noted that nodes on the same ring do not necessarily have any trust relationship with one another, except for the first ring.

The purpose of a matryoshka is twofold: the storage of the user's data in the nodes of the first ring and access to this data in a privacy-preserving manner by other users. The social network application information pertaining to the user associated with the matryoshka is thus replicated on all the nodes of the first (innermost) ring that belong to the trusted contacts of the user. As further described this information will be protected under some encryption. When a message is addressed to a user or when a user attempts to read another user's data, the message or the access request is transferred from a node on the outermost ring of the target user's matryoshka to a trusted node on the inner ring. The message or the request is further forwarded by each node on the path to a node on the next inner ring until it reaches the node representing the target user (the owner of the matryoshka). It should be noted that during the transfer of messages destined to a user, nodes on subsequent rings exchange data with one another (have a link) only if they are trusted contacts of one another in the sense of the social network. Moreover this trust relationship doesnt need to be transitive and nodes on the path to a target user's node do not have to be trusted by the target user. Each user builds her matryoshka the very first time she joins the network, then keeps updating it. Privacy is achieved based on the existing hop-by-hop trust relationships of the social network through the matryoshka structure because it is unfeasible to retrieve a user's trusted contacts while it is feasible to access the user's social network information (profile) or to exchange data with her. Matryoshkas also assure cooperation enforcement based on hop-by-hop trust of the social network as communications only take place among trusted contacts.

2) Peer-to-peer substrate: The peer-to-peer substrate provides global access to a user's data, based on her identifiers. As another view of the system, the peer-to-peer substrate consists of all nodes that are organized in a distributed hash table (DHT) akin to peer-to-peer systems (while in the figures throughout the paper we show a ring-based DHT like Chord [9] for comprehensibility, we plan to use a Kademlia based approach [10] for shorter response times).

ISBN: 978-81-936820-0-5

Series: AIJR Proceedings
Proceedings DOI: 10.21467/proceedings.1

The location of each node on the DHT is determined based on the former's pseudonym and location data is registered according the DHT protocol. The data registered in the DHT to locate a profile thus consists of pointers to nodes on the outermost ring of the requested users's outermost matryoshka.

3) Trusted identification service: The trusted identification service will grant each user and the node thereof a unique pseudonym, a unique node identifier and two certificates for the authentication of the node under each type of identifier. The main purpose of the trusted identification is to prevent Sybil and impersonation attacks and attacks on the DHT overlay. Even though the identification service can be viewed as a centralized infrastructure, the decentralized aspect of the social network system that uses it is not affected since this service's jurisdiction is limited to the purpose of authentication. Moreover this service can be implemented in a decentralized fashion, and provided offline. Furthermore, the hash of a user's name as used in the DHT, her node's pseudonym and node identifier are all uncorrelated in order to prevent simple dictionary attacks.

4) Orchestration: Using the trusted identification service, each user will get a node identifier and a pseudonym for his node in the system together with a certificate associating a different user generated public key for each identifier. The node that represents the user will then be inserted in the peer-to-peer substrate based on the pseudonym. The node will also start its registration by creating secure links with the nodes of all the trusted contacts in the social network. The matryoshka of the user will then be built by initiating registration messages from the core node to the outermost ring with each of the nodes in the innermost ring. Each registration message will eventually reach a node on the peer-to-peer substrate whereby an entry for the user will be created. When a user looks up another user's data in the social network, it search the peer-to-peer substrate for one or a set of nodes that are members of the outermost layer of the the target user's matryoshka. The node in the peer-to-peer substrate that is responsible for the lookup will redirect the request to a node located on the outermost layer of the target user's matryoshka. The request will be forwarded to nodes on inner rings through hop-by-hop trusted links until it reaches one of the nodes on the innermost ring whereby a replica of the social network information of the user is stored. The data will be returned to the requestor through a reverse path across concentric rings of the matryoshka. The actual access to various parts of the information will be controlled based on encryption as explained further in the paper. Figure 2 presents the overall system draft.

B. Core components

In order to implement the privacy preserving social networking application in the described infrastructure, we utilize some well known concepts, as follows.

In order to implement user and data authentication, as well as end-to-end encryption, we employ straight forward public key cryptography. Each node holds a set of properties N, e.g.

the user's full name and birthday, and a proof that it is the owner of these properties. It generates two key pairs: I and P.

The identification service derives the pseudonym P and the node identifier from N and certifies the authenticity of both I and P. The node identifier is used to identify a member of the social networking application, while the pseudonym is used as an identifier in the peer-to-peer system. Possession of pseudonym and node identifier is proved by P and I respectively. I,P and P are not linkeable for any other than the trusted contacts of a user.

Public key cryptography is also used to implement an access control to each attribute of a user's profile. All attributes on the profile of a user are encrypted using a respective private key and the user may choose to share this public key with the selection of other members, whom it wants to share this attribute with. The access control thus is quite basic for the moment. Note however, that in this paper we focus on sketching the new social networking approach and there exists a wealth of group key schemes that we intend to analyse and of which we intend to harness an adequate solution in the future.

C. Operations

In the following we sketch the operations of our system that implement the social networking service, which consist of

- account creation
- profile publication
- data retrieval
- contact request and acceptance
- message management and matryoshka maintenance

1) Account creation:

In order to create an account, the user V has to be invited by a different user U, who is already participating in the system. On invitation, the account is created in four separate steps: 1) identity creation and authentication, 2) joining the P2P substrate, 3) profile and 4) matryoshka creation.

Identity creation:

In order to create its identity, V generates the two key pairs I and P and sends a request to obtain pseudonym, node identifier and certificates from identification service to U that relays the request, addressed to the identification service using the DHT, and on reception provides V with the response. Identification service derives node's pseudonym $Pv = h1(N)$ and its node identifier $v = h2(N)$ from node s properties N, with $h1$, $h2$ being two distinct cryptographic hash functions. It also provides two certificates $\{I^+, v\}S_{TTP}$ and $\{P^+, P_V\}S_{TTP}$ its signature.

Joining the P2P substrate:

On reception of the certifications, V joins the P2P substrate using U as a bootstrapping host and Pv as its pseudonym.

ISBN: 978-81-936820-0-5
Proceedings DOI: 10.21467/proceedings.1

Series: AIJR Proceedings

Creation of the profile:

Unrelated to the certification of its keys, *V* may already create its profile, which contains of atomic attributes for each entry, and creates public key pairs, which it signs with *I*+, for each attribute in order to share it with a subsequently selected group of users. Each attribute is subsequently encrypted with its respective private key. The friend list presents a special attribute, as it contains information on other users. Hence, *V* retrieves the name-attribute from its contacts (*U*, at this stage) in their encrypted form and lists these, finally encrypted with its own respective key, as the friend list. Only a user that is authorized by each of the respective members in consequence is able to access the contained knowledge.

Matryoshka creation:

Finally, *V* creates its initial matryhoshka in order to build the replication service and layers of indirection. As *V* initially only knows *U*, it is solely able to select this node as part of the innermost shell of its matryoshka, for replication and anonymization purposes. It hence stores its encrypted profile with *U* at this point only. Subsequently it sends a request to create a matryoshka path and register itself (cmp. Fig.: 3) to the DHT with its node identifier or a hash of its name as the lookup key for registration6. *V* subsequently sends a registration message and time-to-live counter ttl, to *U*. It signs the registration request, which contains k, the lookup key for the DHT, its own node identifier, the pseudonym of the node on the next matryoshka shell (Pu) and its own public key and node identifier (both certified by the TTP), encrypted to the adressed node on the next matryoshka shell:

$EPu\{Mvu, ttl\}$ with $Mvu = \{k, v, Pu, \{I_v^+, v\}S_{TTP}\}SI_v$

u on reception randomly selects a node from its contact list to be the node w on the next shell of v's matryoshka, and creates a registration request with the pseudonym of w as the next node, encapsulating the whole registration Mvu, and sends it together with the decreased ttl counter ttl, encrypted for w :

$EPw\{Muw, ttl\}$

$Muw=k,Pu,Pw,\{P_u^+,Pu\}STTP,Mvu\}SPu$

The matryoshka path then is created recursively until the ttl, which is initially set to the desired number of shells for the matryoshka, expires. Finally, the node on the outermost shell registers the key and proves its authentication using the chain of encapsulated signatures.

[6]V may, of course choose to hash different parts of the name and probably even other profile attributes to increase its visibility, this, however, does not change the operations fundamentally and hence is disregarded for the rest of this paper.

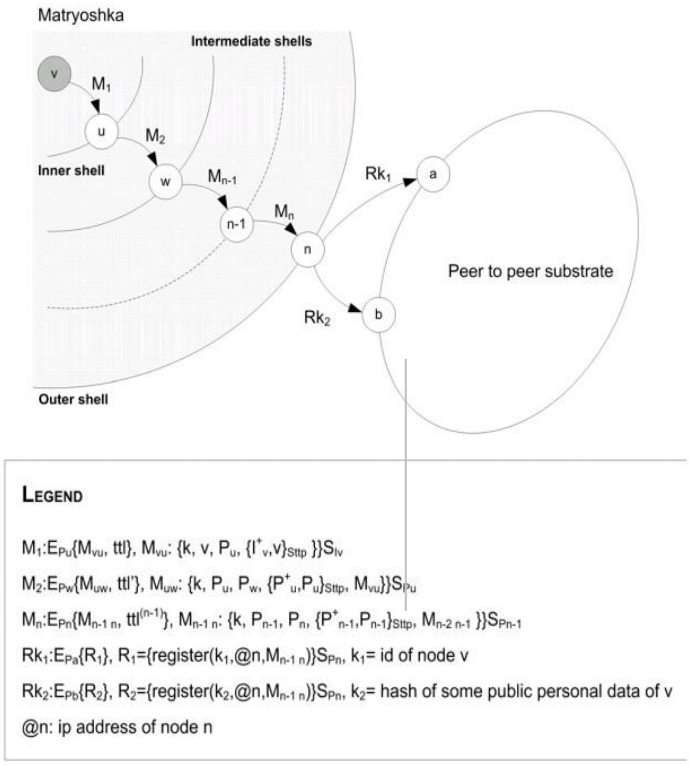

Figure 3: Account and matryoshka creation with key registration for node v.

2) Profile publication: Generally, any data in the system may be public, protected or private. Public data is published by the node and replicated at the trusted contacts of a user that make up the innermost matryoshka shell. Protected data is encrypted by the owner and again replicated at the trusted contacts of the user. All published data, be it public or protected, is signed by the originator. Private data is stored by the owner itself and not published.

Each node may manage three different kinds of data:

1) profile information

2) trusted contact relations

3) messages

The profile information is organized in atomic attributes and consists of the data that a member wants to publish in the social networking service. The trusted contact relations represent a user *s* direct social network and contain all the contacts of a member, each associated with a certain trust level. They hence can be viewed as the friend list of the user. Messages can be exchanged between members of the system and may either contain personal messages or comments to the profile. On reception of messages that contain comments to its

ISBN: 978-81-936820-0-5
Proceedings DOI: 10.21467/proceedings.1

Series: AIJR Proceedings

a profile, a user may either accept them to be annotations, sign them and republish them as either public or protected attributes or simply drop them. The trust level, which is stored with the contact relations, is harnessed in order to select closely related contacts, thus allowing for publication of bulky data like images or maybe even videos, which some rather weak acquaintances of a user may not be interested in replicating.

3) Data retrieval: The lookup of data follows the opposite path compared to the registration. Profile requests are initially routed through the DHT until reaching the registrating node, which replies by sending the registered nodes on the outer shell of the matryoshka. The requesting node then sends its request to the outer shell of a node v'*s* matryoshka, where the lookup is performed recursively. The profile data is approached hop-by-hop through the shells. Reaching a replicating node (or the core of the matryoshka), the request is served and a response sent along the same links through the matryoshka that the request traversed. The node on the outermost shell sends the reply directly to the requesting node. The requesting node then is be able to access the returned data according to its privileges that were assigned by the data'*s* owner through providing the respective keys.

4) Contact request and acceptance: In order for *V* to add another member *U* as a trusted contact, a contact request message is sent in the same way as profile data requests. If *U* accepts the contact relation, it replies by establishing a direct trusted link to *V* and both expand their inner shell of their matryoshka by the other member and performing profile replication and profile registration like in III-C1 (cmp. Fig.:4)

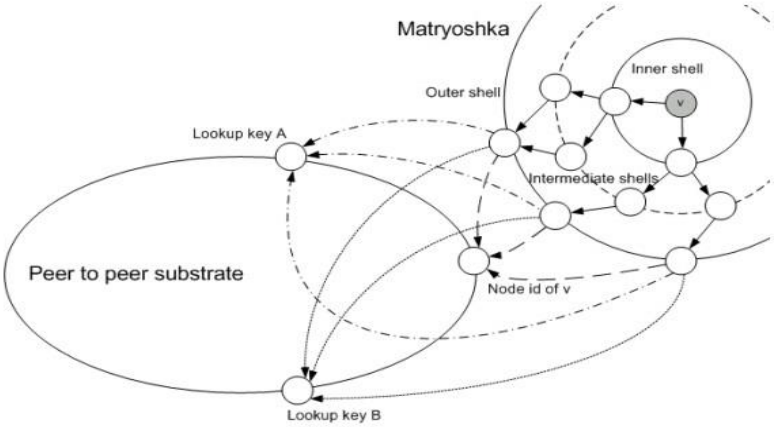

Figure 4: Multiple path creation for node v.

5) Message management: Messages can be viewed as being merely requests to deliver some data. They hence are delivered the same way as data retrieval requests, with the difference that these requests are not served by replicating peers, but by the address, only. Like in the common social networking services, our system is able to deal with two different, public and private,

messages. Public messages contain comments or wallposts that are addressed at the public or probably a selected audience, private messages are messages sent to the addressee personally. While sending a message to a destination node, the sender signs its message with its own private node id key and encrypts it for the receiver, in order to provide integrity and both sender and data authentication. As soon as the receiver gets the message, it can decrypt it and, in case of it being a wall post, publish it as an annotation to its profile.

6) Matryoshka maintenance: If a node that belongs to v'*s* matryoshka leaves the network, it sends a path invalidation message to the nodes on the next inner and outer shells of the matryoshka. The message sent outwards is forwarded to the outermost shell and the nodes deregister themselves from the matryoshka. The node on the previous inner shell will repeat the iterative registration procedure described in III-C1.

In order to account for possibly failing nodes on the paths all nodes regularly check their links in all the matryoshkas they are part of and on detection of failure perform the same updating process.

4 EVALUATION

A complete performance evaluation of the proposed systems is not available at this point, as we are describing work in progress. In the following we however give an overview to evaluate the compliance with respect to the security objectives we required in section II.

Authentication: User authentication in our system is given due to the public key pair all users possess, which are certified by the trusted identification service. Please take note of the fact, that this does break neither the privacy of users, as this TTP does not hold any information about the user, nor the decentralization, as a distributed TTP, based on threshold cryptography, or even multiple TTPs can be used and these do not have to be online at all times. Impersonation attacks are impossible, as in this early approach of our system the users have to prove their identity to the TTP. However, there has been work on mutual authentication in distributed environments without a TTP [11] and we are currently developing further techniques for this purpose.

End-to-End confidentiality: The end-to-end confidentiality is at stake when 1) retrieving profile attributes, 2) sending private messages or 3) commenting on another user *s* profile.

In the proposed approach it is given, as 1) the profile attributes are encrypted using the keys for the corresponding groups by the owner of the profile, and for both 2) and 3) each message exchange is encrypted with the public key I^+ of the address. As the serving peer might not know the requesting peer, man-in-the-middle attacks in principle are feasible, if a node manages to place itself on the path between the requesting and the responding peer. We reserve a special discussion on the possibility of these attacks in our system, below.

Privacy: For each user, none of the group of untrusted parties of the system can gather any information other than attributes that are made public. This holds true for location information of the user, as this is hidden through the anonymization steps through the

ISBN: 978-81-936820-0-5
Proceedings DOI: 10.21467/proceedings.1

Series: AIJR Proceedings

matryoshka, and for the profile information, as it is encrypted and access keys are only given to users that are trusted, and hence belong to one of the groups (e.g. for contact information, business information, a personal blog, pictures, private pictures etc.) managed by the user himself. However, as a peer may serve as a cache in the matryoshka of one or multiple of his contacts, some information about his contact list could be derived from monitoring his traffic or his actions. This does not threaten the members privacy, as the registration message at publishing is the only message in which some identity of the node is disclosed. In this operation as a matter of fact, the node id of the core of the matryoshka and the pseudonym for all other nodes on the path are part of the message. However, since only trusted contacts and the TTP can map the pseudonym to the node id, no untrusted nodes can derive any knowledge from the message. In consequence, nodes on the second innermost shell may infer from the signature chain, that their predecessors in these paths are direct contacts of the core. This does not break the privacy, as all direct links in the matryoshka are trusted.

Controlled access to attributes of the profile: The group based access control scheme, which builds on a group of members that are allowed to see an attribute of the profile, and which is managed locally at the publishing member, allows for a fine grained access control.

Data integrity: Any party can check the integrity of any retrieved (and decryptable) message, as all transmitted data is signed by the originating party (the owner of the respective profile), and as a peer can retrieve the public key, signed by the identification service, at any point in time when the profile information is retrievable as well.

Resilience to impersonation attacks: Due to the existence of the identification service and the authentication by means of public key cryptography, the authentication can be considered as strong as currently possible by means of public key cryptography.

Availability of profiles: The complete profile of a user is replicated and made available at all the members of the innermost layer of his matryoshka, which are all contacts from his contact list, i.e. the peers that are run by his friends and acquaintances in his real life; additionally these replica are accessible from the anonymous peer-to-peer substrate via multiple different paths through the matryoshka. These properties do not guarantee the availability of the complete information of all profiles at any time. However, as the replicating peers and each hop in the matryoshka can be considered as highly trustable, and in consequence as very cooperative, too, we expect to achieve a high availability of data. We discuss the black hole attack, a special attack on the availability, which, due to their decentralized characteristic, is well possible in peer to peer systems, in the subsequent paragraph. An entirely different aspect of availability is the ability of a system to deliver results in an acceptable time. The choice to build the system in a decentralized manner may pose a significant burden to its responsiveness. However, recent studies have shown, that a timely access of data is well possible in peer-to-peer systems [12]. We are in the process of creating a model of the system in order to be able to evaluate

the availability in both its aspects of the responsiveness and the possibility to reach information, in the near future.

Resilience to Man-in-the-Middle and black hole attacks. Both the man-in-the-middle and the black hole attack require the attacker to be able to place himself between the requesting and the responding party. In our system, this in general is possible at three points: by occupying all positions in the DHT which point to the matryoshka of a targeted user, by creating a forged outer shell (through re-registering controlled peers for all keys to a targeted profile) and by placing controlled nodes on one or multiple paths in the matryoshka.

The first strategy is impossible due to the central and trusted assignment of pseudonyms that determine the position of a peer in the DHT. Our scheme to sign requests for registration allows the registering peer to authenticate all requests and hence prevents a malicious peer to forge additional entries to the target'*s* matryoshka. Arbitrarily placing a controlled peer on a path inside a matryoshka would require two steps: 1) to have a trusted contact inside the matryoshka (either as a genuine friend or through an impersonation attack) that 2) needs to collude in selecting the malicious peer as the peer of the next shell, thus possibly inflicting collateral damage on its trusted contact on the shell closer to the core, which actually might be a direct friend of the core itself. Thanks to the identification service an impersonation attack is not possible and as all hops in the matryoshka connect mutually trusted parties, we do not assume the inner layers to mount an attack that would directly damage trusted relationships of its contact in order to achieve 2).

5 CONCLUSION AND FUTUREWORK

In this paper we have analyzed the approach to online social networking that builds on peer-to-peer methods in order to remove a central, omniscient entity. The approach leverages on the external trust that the users have in their friends and acquaintances, both for replicating profiles and for the anonymization of traffic. More precisely the proposed system consists of three main components: an identification service (that can be implemented in a decentralized way and does not have to be online) for certification of public keys and the assignment of pseudonyms; a set of concentric shells around each participant, the matryoshkas, which serve to replicate the profile data and anonymize the traffic for this participant; and a straight forward DHT as a peer-to-peer substrate for the location of matryoshkas in order to be able to access profile data and exchange messages. The proposed system is able to deliver the same services as the existing centralized SNS, but offers stronger privacy protection for its users.

Currently, we can identify a couple of open issues and possible enhancements, which we plan to address in future work. The primary goal has to be to develop a full model of the system, in order to be able to study the availability of data, both with respect to its accessibility and the response times. We see a need to enhance the group and key management scheme, in order to allow for features such as forward or backward secrecy, as well as key revocation without the need to rekey the whole affected group. Another point will be to enable users to reliably

ISBN: 978-81-936820-0-5
Proceedings DOI: 10.21467/proceedings.1

Series: AIJR Proceedings

delete data from the system, which may be easy as long as only the nodes on the innermost shell of the matryoshka offer a replication service but might be increasingly difficult if additional replication services have to different keys for tailored access to selected parts of the profile.

REFERENCES

[1] d. m. boyd and N. B. Ellison, Social network sites: Definition, history, and scholarship. Journal of Computer-Mediated Communication, vol.13(1), 2007.

[2] L. Bilge, T. Strufe, D. Balzarotti, and E. Kirda, All Your Contacts Are Belong to Us: Automated Identity Theft Attacks on Social Networks, 2008, under submission.

[3] S. Moyer and N. Hamiel, Satan is on My Friends List: Attacking Social Networks, www.blackhat.com/html/bh-usa-08/bh-usa-08 archive.html, 2008.

[4] "Sophos Facebook ID Probe," http://www.sophos.com/pressoffice/news/articles/2007/08/facebook.html,2008.

[5] G. Hogben, Security issues and recommendations for online social networks, ENISA Position Paper, 2007.

[6] A. Poller, "Privatsphärenschutz in Soziale-Netzwerke-Plattformen, Fraunhofer SIT Survey, www.sit.fraunhofer.de, 2008.

[7] "Modelling the Real Market Value Of Social Networks, http://www.techcrunch.com/2008/06/23/modeling-the-real-market-value-of-social-networks/, 2008.

[8] J. R. Douceur, The sybil attack, in International Workshop on Peer-to-Peer Systems, 2002, pp. 251 260.

[9] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan, Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications, in ACM Applications, Technologies, Architectures, and Protocols for Computer Communication, September 2001, pp. 149 – 160.

[10] P. Maymounkov and D. Mazieres, "Kademlia: A Peer-to-Peer Information System Based on the XOR Metric," in LNCS: International Workshop on P2P Systems, vol. 2429, 2002, pp. 53 – 65.

[11] B. C. Popescu, B. Crispo, and A. S. Tanenbaum, "Safe and Private Data Sharing with Turtle: Friends Team-Up and Beat the System," in LNCS: Security Protocols, 2006, pp. 213 – 220.

[12] M. Steiner, D. Carra, and E. W. Biersack, Faster Content Access in KAD, in Peer-to-Peer Computing, 2008, pp. 195 204.

[13] J. Li and F. Dabek, "F2F: reliable storage in open networks, in Peer-to-Peer Systems (IPTPS), 2005.

[14] B. W. Ian Clarke, O. Sandberg and T. W. Hong, Freenet: A Distributed Anonymous Information Storage and Retrieval System, in Workshop on Design Issues in Anonymity and Unobservability, 2000, pp. 46 – 66.

[15] K. Bennett and C. Grotho, Gap - practical anonymous networking, in Proceedings of Privacy Enhancing Technologies workshop (PET 2003). Springer-Verlag, LNCS 2760, 2003, pp. 141 160.

[16] P. Matejka, "Security in peer-to-peer networks," Master s thesis, Department of Software Engineering, Charles University, Prague, 2004.

[17] K. P. N. Puttaswamy, A. Sala, and B. Y. Zhao, Improving anonymity using social links, in Secure Network Protocols (NPSec), 2008.

[18] Leucio Antonio Cutillo Refik Molva Thorsten Strufe Institute Eurecom Sophia Antipolis France "Privacy Preserving Social Networking Through Decentralization.