Privacy Preserving Picture Sharing: Enforcing Usage Control in Distributed On-Line Social Networks

Leucio Antonio Cutillo Refik Molva Melek Önen
EURECOM, Sophia-Antipolis, France
{cutillo, molva, onen}@eurecom.fr

Abstract

The problem of usage control, which refers to the control of the data after its publication, is becoming a very challenging problem due to the exponential growth of the number of users involved in content sharing. While the best solution and unfortunately the most expensive one to cope with this particular issue would be to provide a trusted hardware environment for each user, in this paper we address this problem in a confined environment, namely online social networks (OSN), and for the particular picture sharing application. In current OSNs, the owner of an uploaded picture is the only one who can control the access to this particular content and, unfortunately, other users whose faces appear in the same picture cannot set any rule. We propose a preliminary usage control mechanism targeting decentralized peer-to-peer online social networks where control is enforced thanks to the collaboration of a sufficient number of legitimate peers. In this solution, all faces in pictures are automatically obfuscated during their upload to the system and the enforcement of the obfuscation operation is guaranteed thanks to the underlying privacy preserving multi-hop routing protocol. The disclosure of each face depends on the rules the owner of the face sets when she is informed and malicious users can never publish this content in clear even if they have access to it.

Categories and Subject Descriptors K.4.1 [*Public Policy Issues*]: Privacy

Keywords Usage control; picture sharing; Distributed On-Line Social Networks

1. Introduction

The problem of usage control which refers to the control of the data after its publication, is becoming a very challenging problem due to the exponential growth of the number of users involved in content sharing applications. Online social networks like Facebook, Twitter or Linkedin, are becoming the way of communication among people in the Internet. and unfortunately this problem can have a severe impact on privacy in such an environment with several hundreds of millions of registered users. Indeed, even with current privacy protection and access control solutions, users loose their control over data after its very first publication in the network. For example, although a user who uploads or ”posts” some data can indeed prevent unauthorized access to it, she cannot control the further user of it after its publication.

Recently, several peer-to-peer (P2P) based Distributed Online Social Networks (DOSN) have been proposed to preserve users' privacy [4]. In all these solutions, users' data is not stored by a centralized OSN provider anymore. Such a DOSN can be considered as a good candidate for usage control mechanisms since it leverages on the collaboration of the users for any operation including data management and privacy protection. In this paper, we propose to exploit the advantage of the underlying peer-to-peer architecture in DOSNs in order to enforce the control of the usage of some specific content, namely pictures. The proposed mechanism relies on the collaboration of nodes and control is enforced thanks to the forwarding of any packets towards several hops before reaching the final destination. The proposed usage control mechanism is designed over a recently proposed DOSN named as Safebook [1] which overcomes the problem of selfishness by leveraging on the real life social trust relationships among users. The underlying multi-hop forwarding solution can directly be used as a basis for the usage control mechanism.

Section 2 introduces the problem of usage control illustrated by picture sharing applications in online social networks. Section 3 describes the proposed mechanism based on a multi-hop enforcement technique originating from the Safebook DOSN. Finally, the security and efficiency of the proposed protocol are evaluated in section 4.

SNS'12 April 10, 2012, Bern, Switzerland

2. Problem Statement

Usage control for picture sharing in OSN Usage control [9] becomes a mandatory requirement given the very large number of users sharing different types of content. The ideal goal of guaranteeing the control over any type of data in any type of platform seems very difficult to achieve. We address this problem in a confined environment which is OSN and in the context of picture sharing since, as previously mentioned, this application is one of the most popular applications for OSN users.

Current picture sharing tools in online social networks allow users to upload any picture. Access rules to these pictures are defined by the owner of the picture that is the one who uploads it. This user has also some abilities to associate an area of the picture to a label: such a function, namely the ***tag***, can be used to inform other users about their presence in the picture. Unfortunately, if users are not "tagged" in the picture, they will never be aware of these pictures. We assume that each person whose face appears in any picture should decide whether her face in that picture should be disclosed or not and therefore she should define the usage control policy regarding her own face.

Decentralized online social networks As previously mentioned, online social networks severely suffer from the centralized control on users' data and its potential misuse. As an answer, several DOSNs[4] propose to design new applications based on a distributed peer-to-peer architecture. Some of them leverage real life social links to construct a network with trusted peers where the correct execution of any network/application operation depends on users' behavior. In order to achieve a good security degree, these solutions define a threshold for the number of misbehaving users and analyze the trade-off between security and performance based on this degree: for example, in some solutions a packet must pass through a threshold number of nodes before reaching its destination in order to guarantee a certain security degree.

Such peer-to-peer online social networks can be considered as a good candidate for usage control mechanisms in picture sharing. In such an environment, a well behaving node would automatically obfuscate all faces in any picture it receives from other nodes. Therefore, given a threshold number of misbehaving or malicious nodes, in order to guarantee the correct execution of the usage control mechanism, the application can define a minimum number of nodes a legitimate message has to pass through, before reaching its final destination. Among these nodes at least one node should behave legitimately and apply the required protection operations. Additionally to the owner of the picture, only the owner of the face included in that picture should be able to have an initial access to the face in that picture. The further usage control rules for the dedicated face have to be defined by the corresponding user and the correct appliance of these rules should be verified at each node in the path towards the destination.

3. The proposed usage control mechanism

In this section, we describe a usage control mechanism enforced thanks to the cooperation among multiple users that perform multi-hop forwarding. As previously mentioned, the idea of the proposed mechanism is to exploit the distributed nature of peer-to-peer online social networks and to leverage real-life social links to control the access to pictures: as opposed to centralized solutions, all operations are performed with the collaboration of multiple nodes. Thanks to this multi-hop enforcement in this distributed setting, cleartext pictures will only be accessible based on the rules defined by users whose faces figure in the corresponding pictures. Intermediate nodes will detect faces on pictures they receive and verify whether users appearing in the picture have defined any rule on the usage to their face. A social network that answers the previously described requirements is proposed in [1] as a distributed privacy preserving online social network named as *Safebook*. We briefly summarize its characteristics before describing the newly proposed usage control mechanism.

3.1 Safebook: a P2P DOSN leveraging real life trust

The main aim of Safebook is to avoid any centralized control over user data by service providers. The correct execution of different services depends on the trust relationships among nodes which are by definition deduced from real-life social links. Safebook defines for each user a particular structure named as ***Matryoshkas*** ensuring end-to-end confidentiality and providing distributed storage while preserving privacy. As illustrated in figure 1, the Matryoshka of a user $\mathcal{V}$, namely the ***core***, is composed by several nodes organized in concentric shells. Nodes in the first shell, also called ***mirrors***, are the real life friends of $\mathcal{V}$, and store her profile data to guarantee its availability. If a requester $\mathcal{U}$ would have directly contacted one of $\mathcal{V}$'s mirrors, say $\mathcal{A}$, $\mathcal{U}$ would have been able to infer the friendship relation between $\mathcal{V}$ and $\mathcal{A}$. To protect such an information, several multihop paths, ***chains*** of trusted friends, are built where every user's node selects among her own friends one or more next hops that are not yet part of the core's Matryoshka. $\mathcal{A}$ can then be seen as the root of a subtree with branching ***span***[1] whose leaves, namely the ***entrypoints***, lie in the outermost shell.

When a user $\mathcal{U}$ looks for $\mathcal{V}$'s data, her request is served by the entrypoints of $\mathcal{V}$'s Matryoshka and forwarded to the mirrors along these predefined path. The answer follows the same path in the opposite direction.

To prevent malicious users from creating multiple identities, identifiers are granted and certified by the last component of Safebook, the ***Trusted Identification System***(TIS). The TIS is contacted only once during the user registra-

[1] for the sake of clarity, we will consider span=1 in the rest of the paper.

tion phase and does not impact the decentralized nature of Safebook's architecture since it is not involved in any data communication or data management operation.

We now present a new usage control mechanism taking Safebook as a basis.

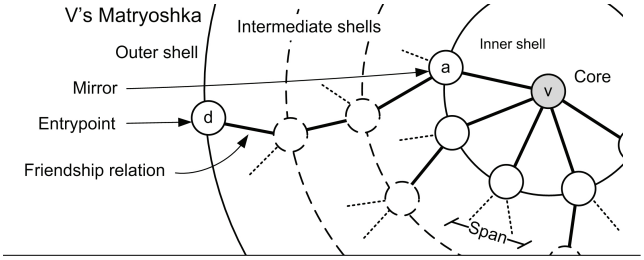

Figure 1. The Matryoshka graph of a user $\mathcal{V}$, from [1]

3.2 Overview of the solution

In the particular environment of Safebook, a user can mainly play three different roles:

- she can **publish** a picture: in this case, she is represented as the core of her own Matryoshka and her friends will store this picture. She also has the option to tag some of her friends who might appear in the picture.
- She can act as a **forwarder** for some pictures: in this case, she belongs to the Matryoshka of either the owner of the picture or the owner of a face tagged in that picture.
- She can also request to **retrieve** some pictures which belong to one of her friends: in this case, she first needs to contact one of the entrypoints of the corresponding core in order to reach that particular user.

Dedicated tasks have been defined for each of these three roles. For example, before publishing a picture, the client has to perform some picture obfuscation operation. Similarly, as a forwarder, the node has to perform some verification operations in order to check whether the picture it is forwarding is correctly "protected" or not. All these tasks will be described in detail in the following section.

In the sequel of the paper, we use the following notation:

- $\mathcal{L}^{\mathcal{P}}$ denotes the regions in a picture $\mathcal{P}$ where a face appears;
- $\mathcal{F}^{\mathcal{P}}$ denotes the set of faces that appear in $\mathcal{P}$;
- $f_{\mathcal{V}}^{\mathcal{P}}$ denotes user $\mathcal{V}$'s face in $\mathcal{P}$;
- $f_{\mathcal{V}}$ denotes $\mathcal{V}$'s face features which are used for face detection algorithms;
- $\{\phi_{\mathcal{V}}^{+}, \phi_{\mathcal{V}}^{-}\}$ denotes user $\mathcal{V}$'s public and private keys, respectively;
- a message M signed using user $\mathcal{V}$'s private key $\phi_{\mathcal{V}}^{-}$ is denoted by $\{M\}_{S_{\phi_{\mathcal{V}}}}$;
- $E_K\{M\}$ denotes the encryption of a message M with the symmetric key K.

3.3 Solution description

In this section, the solution is mainly presented phase by phase. A user first registers to the social network in order to be able to publish a picture. The user can also act as an intermediate node, namely a forwarder node and check whether the content it receives follows the corresponding usage control rules. Finally, the picture retrieval phase is described.

User registration Whenever a user $\mathcal{V}$ registers to Safebook and joins the network, she first generates a pair of public and private keys $\{\phi_{\mathcal{V}}^{-}, \phi_{\mathcal{V}}^{+}\}$ and sends the public key $\phi_{\mathcal{V}}^{+}$ and some samples of her face $f_{\mathcal{V}}$ to an off-line trusted third party, namely the ***Feature Certification Service***(FCS). The FCS generates a certificate for $\mathcal{V}$ denoted by $Cert\left(f_{\mathcal{V}}, \phi_{\mathcal{V}}^{+}\right)$, which proves that the user with some face features $f_{\mathcal{V}}$ owns the public key $\phi_{\mathcal{V}}^{+}$. This face feature certification phase is performed only once and the user does not need to contact the third party anymore.

Picture publication

To make her picture $\mathcal{P}$ available in the network, the publisher $\mathcal{V}$ has to perform the following main tasks:

- *Picture insertion and face detection*: To publish her picture $\mathcal{P}$, the user $\mathcal{V}$ provides $\mathcal{P}$ to her Safebook client. One of the main components of the system consists of the `face detection` mechanism. The face detector aims at finding the presence of faces in the input picture $\mathcal{P}$ and, if this is the case, it returns their location $\mathcal{L}^{\mathcal{P}}$ [2].
- *Picture tagging*: When the face detector derives $\mathcal{L}^{\mathcal{P}}$, the client uses the second component, namely the `face extractor`, which is in charge of copying every face $f_i^{\mathcal{P}} \in \mathcal{F}^{\mathcal{P}}$ detected in $l_i^{\mathcal{P}} \in \mathcal{L}^{\mathcal{P}}$ in a separate file.

  After this extraction task, the publisher $\mathcal{V}$ is asked to tag each face, i.e. to associate every $f_i^{\mathcal{P}}$ to a profile in $\mathcal{V}$'s contact list. If a face $f_{\mathcal{N}}^{\mathcal{P}}$ is tagged with the profile of its owner $\mathcal{N}$, $\mathcal{N}$ receives a copy of the original picture $\mathcal{P}$ [3] and can decide to publish it again on her own profile. Furthermore, the publisher also decides whether her own face can be made available for the network or not.
- *Picture publication*: Once all known faces $f^{\mathcal{P}} \in \mathcal{F}^{\mathcal{P}}$ are tagged, the client can execute its last component which is the `face obfuscator`: The face obfuscator transforms the face location areas $\mathcal{L}^{\mathcal{P}}$ to uninterpretable areas using any human or computer vision algorithm, thus generating an obfuscated picture $\mathcal{OP}$. In our solution, the face obfuscator simply replaces every pixel in $\mathcal{L}^{\mathcal{P}}$ with a black one. The obfuscated picture can thus be seen as the original one with black shapes hiding every detected face. From the resulting obfuscated picture $\mathcal{OP}$, an unambiguous pic-

[2] We assume that face detection algorithms are secure enough. Their design is out of the scope of this paper.

[3] This doesn't violate $\mathcal{V}$'s privacy, as $\mathcal{V}$ aims at making $\mathcal{P}$ publicly available.

ture identifier I is computed as $I^{\mathcal{P}} = h(\mathcal{OP})$, where $h(\cdot)$ denotes a cryptographic hash function. $\mathcal{V}$'s face $f^{\mathcal{P}}_{\mathcal{V}}$ is then signed together with the certificate $Cert\left(f_{\mathcal{V}}, \phi^{+}_{\mathcal{V}}\right)$, the identifier $I^{\mathcal{P}}$, and an expiration time $expTime$.
Finally, $\left\{Cert\left(f_{\mathcal{V}}, \phi^{+}_{\mathcal{V}}\right), I^{\mathcal{P}}, f^{\mathcal{P}}_{\mathcal{V}}, expTime\right\}_{S_{\phi_{\mathcal{V}}}}$ and $\mathcal{OP}$ are published[4].

- *Picture advertisement*: Once advertised by $\mathcal{V}$ about the presence of $\mathcal{P}$, a user $\mathcal{N}$ can control the disclosure of her face $f^{\mathcal{P}}_{\mathcal{N}}$ in that picture.
  $\mathcal{N}$ may decide to make $f^{\mathcal{P}}_{\mathcal{N}}$ publicly available, and publish $\mathcal{OP}$ together with the following signed message:

$$\left\{Cert\left(f_{\mathcal{N}}, \phi^{+}_{\mathcal{N}}\right), I^{\mathcal{P}}, f^{\mathcal{P}}_{\mathcal{N}}, expTime\right\}_{S_{\phi_{\mathcal{N}}}}$$

  If $\mathcal{N}$ wishes to disclose this picture to a subset of its contact list only, she can encrypt the corresponding message with a symmetric key K previously distributed to the dedicated users. In this case, $\mathcal{N}$ will publish $\mathcal{OP}$ together with $E_K\left\{f^{\mathcal{P}}_{\mathcal{N}}\right\}, I^{\mathcal{P}}$.

Forwarding pictures Every intermediate node $\mathcal{T}$ storing or forwarding an obfuscated picture $\mathcal{OP}$ runs by default the face detector and obfuscator components on $\mathcal{OP}$. These tasks ensure the required privacy property in case some clients are manipulated by malicious nodes.
When storing or forwarding a user $\mathcal{V}$'s publicly available face, a legitimate node $\mathcal{T}$ first checks the validity of the signature $S_{\phi_{\mathcal{V}}}$, the expiration time, and the relation between the face features $f_{\mathcal{V}}$ in the certificate and the ones extracted from $f^{\mathcal{P}}_{\mathcal{V}}$. In case of verification failure, $\mathcal{V}$'s face is obfuscated.

Picture retrieval To retrieve $\mathcal{V}$'s pictures, a user $\mathcal{U}$ who is not included in $\mathcal{V}$'s contact list sends a picture request $pctReq$ message to $\mathcal{V}$ and receives a set of identifiers $I^{\mathcal{P}_j}$ related to $\mathcal{V}$'s publicly available pictures $\mathcal{P}_j$. $\mathcal{U}$ then asks for the identifiers she is interested in. For every identifier $I^{\mathcal{P}_j}$ $\mathcal{U}$ retrieves an obfuscated picture $\mathcal{OP}_j$ and the message

$$\left\{Cert\left(f_{\mathcal{V}}, \phi^{+}_{\mathcal{V}}\right), I^{\mathcal{P}_j}, f^{\mathcal{P}_j}_{\mathcal{V}}, expTime_j\right\}_{S_{\phi_{\mathcal{V}}}}$$

containing $\mathcal{V}$'s publicly available face in that picture. When interacting with her friend $\mathcal{N}$, $\mathcal{U}$ sends her a picture request containing some secret s, and receives a list of picture identifiers $I^{\mathcal{P}_j}$ associated to pictures of $\mathcal{N}$, which are either publicly available, or available to those contacts knowing s, only. $\mathcal{U}$ then detects a match in $I^{\mathcal{P}}$ between the identifiers retrieved from $\mathcal{V}$ and $\mathcal{N}$, and, as she previously received $\mathcal{OP}$ from $\mathcal{V}$, she now asks for the missing information $f^{\mathcal{P}}_{\mathcal{N}}$. At the reception of $piRes = \{E_K\left\{f^{\mathcal{P}}_{\mathcal{N}}\right\}, I^{\mathcal{P}}\}$, $\mathcal{U}$ can retrieve $f^{\mathcal{P}}_{\mathcal{N}}$ since she already owns the appropriate decryption key K shared at the friendship establishment with $\mathcal{N}$.

[4] This phase corresponds to the storage of the picture at $\mathcal{V}$'s friend nodes.

4. Evaluation

In this section, we evaluate the proposed mechanism with respect to different security issues such as eavesdropping, unauthorized access or software manipulation attacks. The impact of these attacks is evaluated based on existing social graphs: in September 2005, Facebook published anonymous social graphs of 5 universities in the United States[5]: California Institute of Technology (Calt.), Princeton University (Princ.), Georgetown University (Georg.), University of North Carolina (UNC), Oklahoma University (Okl.). Each graph is represented by an adjacency matrix A whose non diagonal elements a_{ij} are set to one if user $\nu_i \in V$ is a friend of user $\nu_j \in V$, or zero otherwise. As each adjacency matrix is symmetric, the represented social graph is undirected.

Before presenting the evaluation results, we briefly discuss about the feasibility of the proposed usage control mechanism.

Feasibility The feasibility of the proposed usage control mechanism depends on the robustness and speed of the face detection and verification procedures and on the feasibility of the DOSN at its basis, in our case Safebook.

Face detection [12] and face recognition [13] procedures nowadays run in real time in common personal computers. Most of them [5, 7] make an intensive use of Scale Invariant Feature Transform (SIFT) [6], a well known techique used to extract view-invariant representations of 2D objects. Recognition rates of these solutions raise up to 95% in well known databases such as the Olivetti Research Lab (ORL) one [5, 7]. Other techniques can also be used to improve the recognition rate [11] at the expenses of a bigger face feature descriptor.

The proposed usage control scheme does not put any constraints on the face detection and recognition architecture: when the adopted face descriptor is bigger than the face image itself, a reference face image $r_{\mathcal{V}}$ rather than the feature descriptor $f_{\mathcal{V}}$ itself can be certified by the FCS. This change does not have a concrete impact on the time necessary to compare the reference face in the certificate with that one a user wants to make publicly available.

The feasibility of Safebook has been presented in [2]. The study discusses an inherent tradeoff between privacy and performance: on one hand, the number of shells in a user's Matryoshka should be defined as large as possible to enforce privacy in terms of communication obfuscation and protection of the friendship links, but small enough to offer a better performance in terms of delays and reachability; on the other hand, increasing the number of shells after a certain trheshold does not increase the privacy anymore. Such a threshold depends on the social network graph itself [3], more precisely on the number of hops after which a random walk on the social network graph approximates with a pre-defined error its steady state distribution [8]. In this

[5] http://people.maths.ox.ac.uk/ porterm/data/facebook5.zip

case, the endpoint and the startpoint of the random walk are uncorrelated.

Based on the results of the study in [2], we conducted our experiments by setting a number of shells as high as 4. We assume that the number of online nodes correspond to 30% of the total number of users.

Unauthorized picture broadcast Even if malicious users manipulate the underlying software, broadcasting cleartext faces is prevented thanks to the collaborative multi-hop enforcement scheme. Indeed, it is assumed that there is at least one legitimate node which will execute the correct verification operations and the corresponding transformations in order to protect forwarded packets. Nevertheless, Safebook allows the forwarding of encrypted information to a subset of friends; a malicious member $\mathcal{V}$ may exploit this possibility to further send packets to all of its friends. However, $\mathcal{V}$ may need to set-up a virtual server and establish friendship relationships with all users to provide all of them with a picture $\mathcal{P}$. This kind of attacks can be prevented by setting a maximum proper rate on friendship requests. The malicious node would need to design some server advertisement mechanisms using additional out of band information exchange (outside the Safebook network). Furthermore, a malicious node $\mathcal{V}$ may also ask one of her contacts $\mathcal{C}_1$ to manipulate her Safebook client in order to encrypt and republish an unauthorized picture $\mathcal{P}$ at her own profile. $\mathcal{C}_1$, in turn, may ask the same to a malicious contact $\mathcal{C}_2$. If recursively repeated through the social network graph, this attack may disclose $\mathcal{P}$ to all the contacts of every malicious $\mathcal{C}_n$.This attack again requires the manipulation of the OSN client itself and the impact of such an attack would only be important if the number of malicious users is very large. Fortunately, such a massive scale attack may end on the creation of an environment where the adversary may in turn be a victim for her own private data.

Unobfuscated picture forwarding As previously mentioned, the enforcement of the control on the usage of a given picture is based on the collaboration of users and the correct execution of the previously described operations. However, some users can still be malicious and avoid obfuscating some public pictures. To evaluate the impact of misbehavior, we have simulated the process of Matryoshka creation in which the chains leading from the mirrors to the corresponding entrypoints are built. We assume 30% of the nodes is online, and a fraction of them misbehave. We also assume that misbehaving nodes are always online. Two strategies are adopted to select misbehaving nodes: in the first one, R, they are randomly selected; in the second one, C, they are selected between and in the friendlists of all the nodes with higher weight $w_{\mathcal{V}}$ defined as:

$$w_{\mathcal{V}} = d_{\mathcal{V}} * cc_{\mathcal{V}}$$

where $d_{\mathcal{V}}$ represents the degree of node $\mathcal{V}$, i.e. the number of $\mathcal{V}$'s contacts, and $cc_{\mathcal{V}}$ its clustering coefficient, i.e. the ratio between the number of existing links between $\mathcal{V}$'s contacts divided by the number of possible links that could exist. We define a ***compromised chain*** as a chain that is entirely composed by misbehaving nodes.

Table 1 reports the number of compromised chains m_y^x when misbehaving nodes are as high as x%, and strategy y is applied. The average degree d, the average number of chains p a user can buid and the average number of Matryoshka q are also computed for each social graph. With a limited number of misbehaving nodes, e.g. 10%, these nodes should take part in the same chains to challenge the security of system. Nevertheless, with an increased number of misbehaving nodes such as 25%, a significant ratio of chains $\frac{m_y^x}{p}$ gets compromised (up to 83% in the case of Okl.) and the system does not protect the user's privacy anymore.

	d	p	q	m_R^{10}	m_C^{10}	m_R^{25}	m_C^{25}
Calt.	43.3	14.6	58.3	0.25	0.83	6.38	10.16
Princ.	88.9	30.3	126.5	0.33	5.76	17.72	23.23
Georg.	90.4	32.6	130.4	0.43	6.35	16.12	24.91
UNC	84.4	30.8	123.4	0.38	8.40	14.05	24.15
Okl.	102.4	39.9	159.7	0.48	11.06	18.58	33.24

Table 1. Characteristics summary of examined SN graphs.

Data confidentiality and Anonymity Given an obfuscated picture $\mathcal{OP}$, it should be impossible to retrieve any information about users whose depicted faces are not made publicly available. Since by the very design of the Safebook client, there is no way to query the OSN for the identity of the users whose faces are missing, it is, indeed, not possible for an adversary to extract any useful information from an obfuscated picture. Only friends of a user $\mathcal{N}$ can discover this information and retrieve $f_{\mathcal{N}}^{\mathcal{P}}$. Whenever $\mathcal{N}$'s friend $\mathcal{U}$ receives the list of identifiers of the pictures she is allowed to access, she checks the list of picture identifiers in her cache and may find a match for $I^{\mathcal{P}}$. In this case, $\mathcal{U}$ can ask and obtain $f_{\mathcal{N}}^{\mathcal{P}}$, encrypted with a key K she received from $\mathcal{N}$ previously.

Invalid tagging and picture republishing A malicious user $\mathcal{V}$ may associate $\mathcal{N}$'s face $f_{\mathcal{N}}^{\mathcal{P}}$ with her own profile while tagging a picture $\mathcal{P}$. Nevertheless, $\mathcal{V}$ will not manage to make $f_{\mathcal{N}}^{\mathcal{P}}$ publicly available, unless the features of $f_{\mathcal{N}}^{\mathcal{P}}$ are similar enough to that ones in $Cert\left(f_{\mathcal{V}}, \phi_{\mathcal{V}}^{+}\right)$. However, according to the Face Recognition Vendor Test (FRVT) of 2006 [10] the false rejection rate for a false acceptance rate of 0.001 is 0.01 for state-of-the-art face recognition algorithms. A picture $\mathcal{P}$ can be accessed and republished by a third node $\mathcal{Y}$ that does not appear on it. $\mathcal{Y}$ can in fact store in her profile the obfuscated $\mathcal{OP}$ and any publicly available face

$$\left\{Cert\left(f_{\mathcal{X}}, \phi_{\mathcal{X}}^{+}\right), I^{\mathcal{P}}, f_{\mathcal{X}}^{\mathcal{P}}, expTime\right\}_{S_{\phi_{\mathcal{X}}}}$$

for that picture.

Limitations

In order for the intermediate nodes to verify whether the rules are followed or not, the picture should not be encrypted of course(even if there is obfuscation). This security mechanism cannot be implemented over encrypted messages. However, if a malicious user would want to encrypt the picture in order to circumvent the usage control mechanism, only nodes with corresponding decryption keys can have access to these pictures. Such an attack would require an important communication overhead and its impact on the security would not be that important.

Figure 2 summarizes the characteristics of the proposed solution.

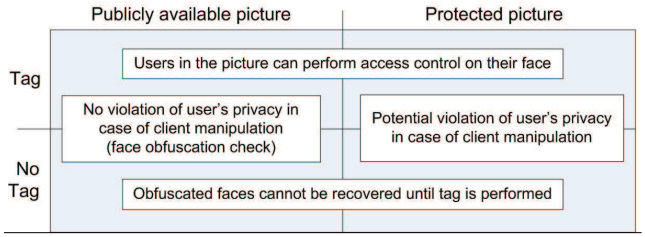

Figure 2. Spread of information vs usage control.

5. Conclusion and Future Work

As it is not feasible to design a perfect usage control mechanism to control the management of any type of data in any environment, we proposed a preliminary solution dedicated to picture sharing tools widely used in the context of online social networks. Although it might be feasible to design such a mechanism in a centralized environment, current OSN providers are not yet interested in such protection mechanisms. On the contrary, decentralized, P2P based online social networks rely on the collaboration of users for any operation including data management and security. The proposed usage control mechanism takes advantage of this inherent cooperation between users and ensures the enforcement on the control of the pictures thanks to a dedicated multihop picture forwarding protocol. The message has to follow a dedicated path of sufficient intermediate nodes which perform the dedicated tasks defined in the usage control policy before reaching its final destination. Thanks to this multihop enforcement mechanism, users whose face appears in a given picture will be able to control its usage in the very beginning stage of its publication. Nevertheless, the protection of the picture and the enforcement of this control is only efficient in the confined environment of the DOSN and when pictures are not encrypted; however, the impact of attacks launched outside this environment or aiming at encrypting the message is very limited within the DOSN.

In the future work, we plan to evaluate the scalability and the performance impact of the proposed solution, and integrate its features in the current Safebook prototype[6].

[6] http://www.safebook.eu/home.php?content=prototype

Acknowledgments

This work has been supported by the RECOGNITION project, grant agreement number 257756, funded by the EC Seventh Framework Programme Theme FP7-ICT-2009 8.5 for Self-Awareness in Autonomic Systems.

References

[1] L. A. Cutillo, R. Molva, and T. Strufe. Safebook : a privacy preserving online social network leveraging on real-life trust. *IEEE Comm. Mag., Consumer Comm. and Networking*, 2009.

[2] L. A. Cutillo, R. Molva, and M. Önen. Performance and Privacy Trade-off in Peer-to-Peer On-line Social Networks. 2010. Technical Report RR10244.

[3] L. A. Cutillo, R. Molva, and M. Önen. Analysis of privacy in online social networks from the graph theory perspective. In *IEEE Globecom 2011, Selected Areas in Communications Symposium, Social Networks Track*, 2011.

[4] A. Datta, S. Buchegger, L.-H. Vu, T. Strufe, and K. Rzadca. Decentralized online social networks. In B. Furht, editor, *Handbook of Social Network Technologies and Applications*. Springer US, 2010.

[5] C. Geng and X. Jiang. Face recognition using sift features. In *Proceedings of the 16th IEEE international conference on Image processing*, pages 3277–3280. IEEE Press, 2009.

[6] D. G. Lowe. Distinctive image features from scale-invariant keypoints, 2003.

[7] A. Majumdar and R. K. Ward. Discriminative SIFT Features for Face Recognition. In *Proc. of Canadian Conference on Electrical and Computer Engineering, 2009*, 2009.

[8] M. Mitzenmacher and E. Upfal. *Probability and Computing : Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, January 2005.

[9] J. Park and R. Sandhu. Towards usage control models: beyond traditional access control. In *SACMAT '02*. ACM, 2002.

[10] P. J. Phillips, W. T. Scruggs, A. J. O'Toole, P. J. Flynn, K. W. Bowyer, C. L. Schott, and M. Sharpe. FRVT 2006 and ICE 2006 Large-Scale Results. Technical Report NISTIR 7408.

[11] C. Velardo and J.-L. Dugelay. Face recognition with DAISY descriptors. In *MM'10 and Sec'10, ACM SIGMM Multimedia and Security Workshop, Rome, Italy*, 09 2010.

[12] P. Viola and M. J. Jones. Robust real-time face detection. *Int. J. Comput. Vision*, May 2004.

[13] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition: A literature survey, 2000.