A Peer-to-Peer Agent Community for Digital Oblivion in Online Social Networks

Klara Stokes and Niklas Carlsson
Department of Computer and Information Science
University of Linköping
Linköping, Sweden
Email: name.surname@liu.se

***Abstract*—We design a system that provides *digital oblivion* for users of online social networks. Participants form a peer-based agent community, which agree on protecting the privacy of individuals who request images to be forgotten. The system distributes and maintains up-to-date information on oblivion requests, and implements a filtering functionality when accessing an underlying online social network. We describe digital oblivion in terms of authentication of user-to-content relations and identify two user-to-content relations that are particularly relevant for digital oblivion. Finally, we design a family of protocols that provide digital oblivion with respect to these user-to-content relations, within the community that are implementing the protocol. Our protocols leverage a combination of digital signatures, watermarking, image tags, and trust management. No collaboration is required from the social network provider, although the system could also be incorporated as a standard feature of the social network.**

Keywords*—Digital oblivion, right to be forgotten, trust, perceptual hash, watermarking, tag, facial recognition, user-to-content relation.

I. Introduction

Online social networks (OSN) are an increasingly important media for publication and communication. The discussion in this paper focuses on Facebook, the most popular OSN, which recently reached one billion active users [16], but the results that we present are also applicable to other OSN. Typically, every user account in an OSN has a "profile", lately transformed into a "timeline", which presents posts and content that are related to the user, in the latter case ordered according to their respective timestamps. In addition to content uploaded by the user herself, the timeline also includes tags (links) pointing to the user from content on other users' timeline.

Facebook supports the removal of unwanted tags. Facebook's privacy settings also allow the user to request that she should review and approve any tags before they are published. When the user removes a tag in Facebook, she is also asked if she wants the tagged content removed, and the reason for this request. Currently, in this situation, Facebook typically only removes content that explicitly violates the Facebook terms. An OSN administrator has to manually evaluate every removal request, which takes time. Typically, the OSN will try to remove the content within 72 hours. If the content does not violate the terms of the OSN, the users only choice may be to directly ask the uploader to remove it. This can be problematic and tough for the victim in many ways. The uploader may refuse to remove the content, it can be embarrassing to ask the uploader directly, etc. In cases of bullying or harassments, Facebook also offers a "Social reporting" tool, which allows the user to share the content that makes her uncomfortable with someone she trusts; e.g., a parent or a teacher. However, as exemplified by recent events, the provided solutions still seem to come up short in practice.

Example 1 (Cyberbullying caused by a photograph)**.** *Amanda Todd killed herself at the age of 15 after repeated cyber-bullying [17]. In seventh grade, she had contact with a 30-year-old man in a chat room who persuaded her to show her breasts for him. Later, he contacted her on Facebook, and gave her an ultimatum: either she made a "show" for him or he would send the photo to everyone she knew. She refused! Later the police knocked on her door telling her that the man had sent the photo to everyone she knew. After this she was a victim for bullying in school and she tried to move. But the man followed her over the Internet, and put up a Facebook page with her bare breasts as profile image, from which he contacted her new friends. She was again a victim for bullying, physical and psycological, and again she changed school, but the cyberbullying was impossible to stop. Eventually she did not see any other solution than suicide. Just before she ended her life, she tells her story in a Youtube video, holding up handwritten notes [2]. One of the notes says: "I can never get that photo back, it's out there forever."*

In this article we present a system that help people in similar situations as Amanda's. While our solution would not allow the photo to be returned to Amanda, the solution provides a way to access Facebook without being constantly confronted with images that the user would like to be forgotten. In a statement against bullying, students (in Amanda's school, for example) could choose to socialize using our system, without reminders of the forgotten content. To further improve their environment, schools could actively incorporate or promote the use of our system as part of their bullying prevention program.

While the outcome is not always as tragic, events such as the one described in Example 1 are common. In fact, many teenagers upload or send explicit photos of themselves to others. This is known as "sexting". Statistics indicate that of female teenagers, 17.3% have sent an explicit photo to someone else, and 30.9% have received such a photo. For male teenagers, the corresponding numbers are 18.3% and 49.7%. It is believed that the difference between the number of teenagers

NOTICE: This is the authors' version of a work that was published in the conference proceedings of the PST 2013 and the IEEE Xplore Digital Library.

who have sent and who have received explicit photos is caused by forwarding. There are at least two documented cases of teenage girls who committed suicide because of an explicit photo they shared with a boyfriend or a flirt, and which later was forwarded to others [14].

During the past few years, there has been an intense discussion in Europe on what is called the "right to be forgotten". In the context of OSN, the right to be forgotten is about controlling data that already is out on the network, providing the possiblity to remove it, or limiting access to it, when so required. There is an endless list of examples of personal problems and tragedies that could have been solved through the right to be forgotten [12]. Previous examples showed several suicides by teenage girls. In December 2012, there were riots at a college in Gothenburg, Sweden, caused by hundreds of photos of nude or half-nude schoolgirls uploaded to an account at Instagram [15].

In this article we present a system that can provide the right to be forgotten in OSN for the described scenarios. The system can be used as part of bullying and anti-harassment protection programs. Our protocols let the users within a community filter out forgotten material from the social network, and leverage a combination of digital signatures, watermarking, image tags, and trust management. No collaboration is required from the social network provider, although the system could also be incorporated as a standard feature of the social network.

A. *State of the Art*

The current opinion seems to suggest that feasible solutions for the right to be forgotten for today's Internet should use legal measures. The 25th of January 2012 the European Commission presented a proposal that included the introduction of the right to be forgotten in the European data protection regulations [6]. The proposed regulation is still to be adopted by the European Parliament.

The proposed regulation has received sharp criticism for the possible negative effects on free speech [12]. It imposes on the OSN provider to take actions whenever a user demands the removal of some data containing information relating to her and the demand cannot be argued to go against the freedom of speech. Otherwise there will be court actions with "a fine up to 1,000,000 euros or up to two percent of Facebooks annual worldwide income" [12]. The simplest solution for the OSN provider may therefore be to simply remove any reported material, a strategy that in general most certainly threatens the right to free speach.

In general, it is problematic to determine who has the right to decide what shall be removed from publication. If I upload some data (e.g. a photo), it can be argued that I should be able to remove it again. However social networks do not necessarily apply this policy. And what if a friend of mine downloaded it first and then uploaded it again, on her own timeline? To whom belongs the right to determine the visibility of the data then? The third possible scenario, in which the first person to publish the data is a friend of mine, is even more complicated.

We will use the notation digital oblivion to denote technical solutions for the right to be forgotten. Among the proposed solutions for digital oblivion, none gives more than a partial protection against unauthorized use of "forgotten" material. Many solutions focus on attaching an expiration date on the published material [7], [8]. The advantage with this approach is that there is no need for the user to be actively involved in the removal of content, nor is there a need for the user to search for material that contain their personal information and that they might want to "forget".

In general, there exist two approaches for implementing digital oblivion with expiration dates in the current literature, one which relies on cryptography, employing for example keys with a date of expiration, and another in which the material is kept on an external, dedicated, trusted server. The latter approach suffers from obvious scalability problems. None of the existing solutions will protect material that was released or copied before the date of expiration.

Digital rights management (DRM) has also been proposed for digital oblivion [8]. For example, in [7], a protocol was proposed that works so that the material is marked with a date of expiration when it is published, and the material is embedded with a fingerprint in subsequent distribution, which later allow for the identification of the user who distributed the material after the expiration date. For more details on the state of the art for digital oblivion we refer to the recent survey paper [8].

In contrast to the above expiry-based protocols, in this article we take a pro-active approach and allow users to forget material that the user have found on the OSN, either through casual surfing, notifications by a friends, or through tagging, for example.

B. *Research Challenges*

On the current research challenges in the area, we cite the conclusions of a recent EU report [8]:

"The fundamental technical challenge in enforcing the right to be forgotten lies in

(i) allowing a person to identify and locate personal data items stored about them;
(ii) tracking all copies of an item and all copies of information derived from the data item;
(iii) determining whether a person has the right to request removal of a data item; and,
(iv) effecting the erasure or removal of all exact or derived copies of the item in the case where an authorized person exercises the right."

This paper addresses challenges (iii) and (iv), under certain assumptions. While solutions to challenges (i) and (ii) could further improve our solution, at the moment we adopt the old adage "what you don't know won't hurt you." Indeed, it can be argued that the scenarios calling for a right to be forgotten that we consider, primarily occur once the disturbing personal data items have been identified and located.

C. *Contribution*

In this article we present a solution that will give users of OSN a restricted functionality of digital oblivion. Compared to the functionalities currently offered by Facebook, our system offers speed and autonomy, meaning that a group of users together can implement the system without the collaboration from the OSN.

The solution can be described as a distributed, user-managed system for access control of content in the underlying OSN, based on authenticable user-to-content relations. The main difference between our system and other systems for access control in OSN, is that we give the user access control also over content that is located on other users' timelines. The system is based on an agent community, with software agents installed by the users. The agents communicate and negotiate in order to agree on what content should be forgotten. The software restricts and filters the user's view of the OSN, ensuring that forgotten content is made invisible to the users who run the agent.

In order to show the feasibility of the solution, we outline a candidate design as proof of concept. We note that parts of our solution can be used separately, in other systems.

- We introduce the idea of a P2P community of agents that can provide a platform for the implementation of collaborative security and privacy solutions. The agent controls the execution of the OSN client, and is therefore an example of specially dedicated software that can provide a functionality of digital oblivion.
- We describe how digital signatures combined with watermarking can be used for digital oblivion. Watermarking can be particularly useful when the user wants to reclaim images that she originally uploaded, and others later uploaded again.
- We describe alternative methods for content uploaded by others. For images, we propose tags as indicators of presence of personal information and provide a protocol based on trust management that delivers digital oblivion in this context. We also describe how, instead of tags, facial recognition can be used for images, and semantics for textual content.

II. A Peer-to-Peer Agent Community for Digital Oblivion

A. Design Goals and Guiding Scenarios

We will consider three distinct scenarios that shall illustrate what is expected by digital oblivion:

- **Scenario 1.** *The user wants to forget material she originally uploaded, appearing on her own timeline.*
- **Scenario 2.** *The user wants to forget material she originally uploaded, now appearing on someone else's timeline.*
- **Scenario 3.** *The user wants to forget material in which she appears, but which was not originally uploaded by her.*

In Example 1, Amanda uploaded the photo herself, so it is an example of Scenario 2. Also the two other suicides reported in [14] are examples of Scenario 2.

The following more generic example introduces several issues that were not present in Example 1 and is designed to illustrate Scenario 3.

Example 2. *Suppose that U attends an event together with another person V, and that at some point a third person W takes a photo of U and V together. Then W uploads this photo to an OSN, without the permission of U.*

Several questions arise.

(i) Suppose U wants to forget the photo, but either V, or W, or both, disagree and insist on that the photo should stay public. Who should decide?

(ii) Suppose that it is clear who should decide if the photo should be forgotten, depending on the role a person has with respect to that photo. For example, if the person is present on the photo, or if she took the photo and uploaded it. Then how do we verify that a particular individual has the role she claims to have?

Example 2 shows that an analysis of *user-to-content relations* (U2C relations) is critical for the correct design of a system providing digital oblivion. The designers should answer at least the following two questions:

(i) Which U2C relations should give the user the right to decide that the content should be forgotten?

(ii) How can these U2C relation be verified in a secure and automatic way?

In our baseline design, we will assume that an individual has the right to decide that some content should be forgotten if there is a relevant U2C relation that can be verified. As a consequence, if two people argue differently, and both holds the right to decide, the person who argues that the material should be forgotten will always win. This seems to be in line with suggestions from the European commission [6], although we agree on that this policy can be critized.

The definition of the problem in terms of U2C relations may be compared with the OSN relations used by Cheng et al. [5] for defining access control policies in OSN: user-to-user, user-to-resource, resource-to-resource relations. However, in this article we use U2C relations for access control on data that traditionally are not within the jurisdiction of the user.

We can use U2C relations to represent the essential difference between Scenario 1 and 2 on the one hand, and Scenario 3 on the other hand. Consider the following U2C relations:

- **U2C R1.** *User u uploaded content c to the OSN.*
- **U2C R2.** *There is personal information on user u present in content c.*

U2C R1 is the relationship type relevant in Scenario 1 and 2. In Scenario 3, U2C R2 also applies. Scenario 3 is more general, and also more difficult. In this article we focus on automatic and secure verification of the authenticity of U2C relations.

B. System Design and Implementation

Let some users of a social network install a software agent with the following properties:

1 *Communication:* The agents of distinct users can communicate over a P2P overlay network.

2 *Filtering:* The agent is capable of (i) intercepting and modifying the material that the user uploads to the OSN, and (ii) deciding what the OSN client will show to the user.
3 *U2C authentication:* The agent community is capable of establishing a protocol that allows for the authentication of some U2C relation.

Then the users can obtain a functionality of digital oblivion with respect to the U2C relation in question.

The P2P community of agents creates a virtual environment within the OSN that will allow the users to claim digital oblivion of already published content. The content will then be removed from this virtual environment, and so from the OSN, as observed by the users within the community. The virtual environment works as a filter, instructing the OSN client to ignore the "forgotten" material. This ensures that the perception of all users who are running the agent is that the "forgotten" material is removed even when it is not physically removed by the OSN. In the meantime additional actions can be taken to request that the OSN provider completely removes the content.

One limitation of our solution is that users that do not install the agent will still be able to see the "forgotten" material. This makes the system unsuitable to protect against undesirable content viewing by people that actively searches for compromising material. However, we note that in many situations the user primarily wants to avoid the daily exposure of compromising material, and/or comments from others about the same. This is in particular the case when we are dealing with bullying or harassments.

In principle, the software agent could be installed and run on voluntary basis. Example scenarios in which the OSN users would be interested in installing the agent may include:

- When the users of the OSN to quickly wants to remove annoying content with personal information, and the content is located on some other user's timeline. This opportunity may provide strong incentive for users to install the agent, and such feature is likely to be used on a regular basis by some people.
- If the use of the agent was visible to the OSN friends of the user, then using the digital oblivion functionality could be an ethical statement. Such statements can be important for users of OSN, as they can help build a positive online personality.
- The functionality could be installed within the OSN. This would give added value to the OSN in question, in terms of user satisfaction.
- Some organizations often confront serious cyber-bullying problems, in particular schools. Such organizations could include the use of the digital oblivion functionality as part of their bullying prevention program. If a school or university maintains one of Facebook's Group for Schools, digital oblivion could be a requirement for joining the school online community.

1) Communication and Filtering: Property 1 (communication) can be easily implemented for devices with public IP address and there are work-arounds for the rest of the cases. We will assume that the topology of the P2P community will be built upon the topology of the existing network structure of the OSN, so that the agents of two friends in the OSN are neighbors in the P2P network.

There are several ways to achieve Property 2 (filtering), including solutions involving server-side and client-side execution. What solution to choose depends on the assumptions made on the liability of the OSN, and specific considerations in implementation, which may vary between different OSN.

Pure server-side execution would allow interception, modification, and filtering of content to be fully incorporated within the OSN. An OSN featuring digital oblivion will provide added value to some users, who may select that OSN before others. Some functionalities of our system could be implemented as an application ("app"), running on the OSN server.

For client-side execution, the agent could be implemented as an application wrapper or a browser plugin. On some systems, like some mobile platforms, there are obstructions for building application wrappers. In this case, maybe the simplest solution is to implement a new OSN client featuring digital oblivion. While our general design is applicable to both server-side and client-side execution, here we present a P2P-based solution that implements client-side filtering.

2) User-to-Content Authentication: From a cryptographic perspective, perhaps the most interesting required property in our system is Property 3: distributed authentication of U2C relations. Here, we propose methods that can offer authentication for the two U2C relations that we introduced in Section II-A.

In the case of U2C R1, we propose a protocol that achieves U2C authentication through a combination of digital signature and watermarking techniques. The protocol is based on the combined digital signature and digital watermark scheme presented in [4], which allows a digital signature to be embedded in the image using a watermark scheme. Although their motivation was to reduce the extra bandwidth that is typically required when attaching a digital signatures, we use their solution for another purpose. In our context, the extra bandwidth is not the concern, however it is important that the digital signature cannot be detached from the image. By using a combined scheme similar to the one in [4], we achieve this property. Any agent in the community can then verify the U2C R1, by retrieving an hash from the image in two ways, (i) from the digital signature in the watermark embedded in the data (employing the public key of the agent who uploaded the content) and (ii) by extracting the hash directly from the image, and subsequently compare the two signatures.

In the case of U2C R2, we propose three different methods to detect indications of personal information in content. (1) *Tags* can indicate presence of personal information in OSN images. According to Facebook: "A tag is a special kind of link. When you tag someone, you create a link to their timeline." So a tag is link between an image and an individual OSN account. Tags are often used in comments about persons appearing in images. The main motivation for using tags in U2C R2 authentication is that the presence of personal information is explicitly confirmed by a user who has no interest, or negative interest, in the U2C R2 authentication. (2) *Faces* also indicate presence of personal information in images. Facial recognition can be used to find specific faces in images automatically. Indeed, Facebook uses facial recognition to suggest tags to the OSN users [10]. (3) For textual content, probably the best way to find indications of personal information is to use *semantics* [3], and other tools from natural language processing [1].

Alone, these indicators of presence of personal information in content are too weak to provide secure U2C R2 authentication. Tags can be added with the intention to falsely indicate personal information where it is not present. Facial recognition can indicate the presence of a certain face in an image, but the information typically requires confirmation feedback from humans, and the same is true for semantics in textual content. Security can be added by using trust management in the agent community, as in the protocol that we propose.

III. Protocols

Here we present a protocol suite that implements digital oblivion in a candidate design of our system.

A. Dependencies

The described protocols make use of the following four building blocks.

A public-key signature scheme: We recommend to use an anonymous signature scheme, that is, an eavesdropper with access to the signed content should not be able to easily tell who published the material, as long as the owner still did not post a demand for its removal. The public-key signature scheme used by our protocols requires three basic functions.

- KeyGen(π): Given a system parameter π, it returns a pair of public and private keys (K_{pub}, K_{priv}).
- Sign(H,K_{priv}): Given a hash H and a private key K_{priv}, it returns a signed hash H_{priv}.
- Verify(H_{priv},K_{pub}): Given a signed hash H_{priv} and the public key K_{pub}, it returns the hash H.

A blind, robust watermarking scheme: A watermarking scheme is blind if the original file F is not needed to recover an embedded string M from the watermarked version Y. It is robust if it is hard to remove M without destroying F. The watermarking scheme used by our protocols requires two basic functions.

- Embed(M,F): Given a string M and a file F, it returns Y which embeds M into F.
- Recover(Y): Given a file Y with an embedded watermark M, it returns the embedded string M.

A robust perceptual hashing function: We use a perceptual hashing function, which should be robust with respect to content-preserving modifications and result in few collisions. See for example [9].

- $H(F)$: Given an image F, it returns a hash H of F.

A trust combination function for the agent community: We leverage existing trust-management protocols to extend our design to allow for disapproval of oblivion requests [11], [13]. For this purpose, we use two trust functions which returns normalized trust values between 0 and 1.

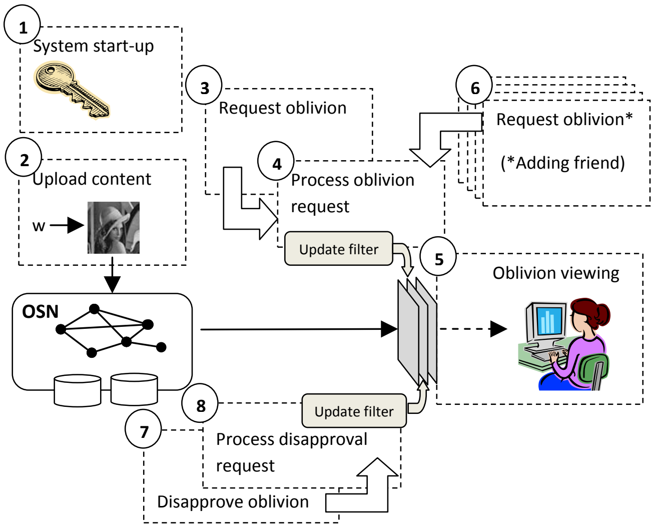

Fig. 1. High-level system overview.

- Trust(A, B): Given two agents A and B, it calculates the trust that agent A has in agent B, using a normalized trust metric.
- TrustComb($A, B_1, \ldots, B_n$): Given agent A and an array of agents $B_1, \ldots, B_n$, this function combines the trust values the agent A has in the agents $B_1, \ldots, B_n$ together. The combined trust can take into account that different agents can have distinct roles and can weigh their trust according to these roles.

B. A Protocol Suite for Community-Based Digital Oblivion

In this section, we present a family of algorithms that together will provide a functionality of digital oblivion. Figure 1 provides a high-level overview of the system and the protocols.

The functionality of each algorithm is here described within its correct context, giving an overview of the protocol suite:

1) *System start-up.* When an agent A joins the community, a trusted dealer runs Algorithm 1, which simply generates a pair of keys for the digital signature scheme and stores an identifier of the agent together with these keys.
2) *Upload of content.* Before the user uploads any content, the agent embeds a watermark in the content, which later will allow the user to claim ownership. This process is described by Algorithm 2.
3) *Request for oblivion of content.* When the user wants to request the oblivion of content, she tells her agent to run Algorithm 3, which will send a message to the neighbors of the agents in the P2P network in limited broadcasting.
4) *Receiving oblivion request.* When an agent in the community receives a request for oblivion of content, it runs Algorithm 4. This algorithm finds the content in question, and if the request can be authorized by

a verified U2C relation, it indexes the content on a list of forgotten content maintained by the agent: the agent's oblivion list. The U2C relation authentication must be done by each agent individually. For this purpose, each agent maintains its own oblivion list. To reduce the size of the list, the agent must only include content that appear on the the friends timelines. Natural extensions and generalizations are of course possible. For example, the list could easily be extended to include also content on the timelines of friends of friends, or some other set which could be made to match the user's privacy settings.

5) *Oblivion viewing.* Before a content is shown by the OSN client to the user, the agent runs Algorithm 5. This algorithm checks if the content is on the agents list of forgotten content. If it is, then the agent instructs the OSN client to ignore the content.
6) *Add a new friend.* When a user adds a new friend in the OSN, the agent will run Algorithm 6, which will send a list of the user's current requests for digital oblivion to the agent of the new friend, as one of more messages. Clearly the number of messages can be reduced by aggregating multiple oblivion requests into a single message.
7) *Disapproval of oblivion of content* When a user does not agree on the oblivion of content, she will tell her agent to run Algorithm 7, which will broadcast a disapproval of oblivion of content message to the community.
8) *Receiving disapproval of oblivion* The agent that receives a disapproval of oblivion of content, runs Algorithm 8, which will evaluate the reputation of the agents involved in order to decide whether to keep the content on the oblivion list or not. Then the trust values of the involved agents are also updated.

Algorithm 1 System start-up

```
$(K_{pub}, K_{priv}) :=$ Keygen$(\pi)$
Dealer stores $Id_A, K_{pub}$
Agent $A$ stores $K_{priv}$
```

Algorithm 2 Agent A uploads content F to the OSN

```
Create a perceptual hash $H := H(F)$ of $F$
Use the digital signature algorithm and $A$'s private key to sign the hash, $S :=$ Sign$(H, K_{priv})$
Create a serial number $N := N(F)$ for $F$
Use watermark embedding algorithm to embed the signature and the serial number into $F$, $Y :=$Embed$(\langle N, S\rangle, F)$
Forward the watermarked content $Y$ to the OSN client, who can upload it to the OSN
```

IV. System Analysis

Here we analyze the security and privacy of the following components of the system: (i) the digital oblivion community, (ii) U2C R1 authentication through digital signature and watermarking, and (iii) U2C R2 authentication through tags and trust management.

Algorithm 3 Agent A requests for oblivion of content F

```
if agent $A$ originally uploaded $F$ then
    $M := \langle N(F), @F, Id_A\rangle$, where $@F$ is a link to $F$
else if user $U(A)$ is tagged with tag $T := T(F, U(A), U(B))$ in content $F$ by user with agent $B$ then
    $M := \langle H(F), T, Id_A, Id_B\rangle$
else if user $U(A)$ receives tag request $T_R := T_R(F, U(A), U(B))$ in content $F$ by user with agent $B$ then
    $M := \langle H(F), T_R, Id_A, Id_B\rangle$
end if
Send $M$ to the neigbors of $A$ in the P2P community through limited broadcasting
Add $M$ to a list of sent oblivion requests $L_M$
```

Algorithm 4 Agent X receives oblivion request M

```
if $M = \langle N(F), @F, Id_A\rangle$ then
    Get content $F$ from link $@F$
    Get public key $K_{pub}$ of the agent with identity $Id_A$ from the dealer
    Use algorithm Recover$(Y)$ to recover the embedded watermark $M$ containing the signature $Z :=$ Sign$(H(F), K_{priv})$
    Use algorithm Verify$(Z, K_{pub})$ to obtain $H(F)$ from the watermark
    Obtain the perceptual hash $H(F)$ directly from the $F$
    Compare the two versions of perceptual hash to see if the signature is valid
    if signature is valid then
        Add $N(F)$ to the list of accepted oblivion requests $L_O$
    end if
else if $M = \langle H(F), T, Id_A, Id_B\rangle$ then
    if $T$ exists and the combined trust TrustComb$(X, A, B)$ is high enough then
        Add $(H(F), B)$ to the list of accepted oblivion requests $L_O$
    end if
else if $M = \langle H(F), T_R, Id_A, Id_B\rangle$ then
    if combined trust TrustComb$(X, A, B)$ is high enough then
        Add $(H(F), B)$ to the list of accepted oblivion requests $L_O$
    end if
end if
```

Algorithm 5 Oblivion viewing of content F

```
if content is watermarked then
    if embedded serial number $N(F)$ is among the serial numbers on oblivion list $L_O$ then
        Instruct the OSN client to forget $F$
    end if
else if perceptual hash $H(F)$ is among the perceptual hashes on oblivion list $L_O$ then
    Instruct the OSN client to forget $F$
end if
```

Algorithm 6 Agent A adds new friend that has agent B

```
for $M$ on $A$'s list of sent oblivion requests $L_M$ do
    Send $M$ to $B$
end for
```

Algorithm 7 Agent C disapproves the oblivion of content F

```
Broadcast the message $\langle H(F), Id_C \rangle$ to the community
```

A. The Digital Oblivion Community

When designing a community-based solution, it is important to note that several privacy issues arise.

- Our implementation of digital oblivion is based on the distributed storage of lists that indexes content that have been requested to be forgotten. There exists an obvious risk that there is a curious user within the community, or some malicious software that forwards oblivion lists to an adversary, who then uses it to identify embarrassing/hurtful content referring to the users in the community. This risk could be mitigated through secure implementation, e.g. encryption of stored data.
- Digital oblivion requires that the user tells the system what data she wants to forget. The use of a distributed system implies that the user has to post her requests to the other users, who consequently will be able to find out what data she wants to forget. This issue could potentially be solved through the use of anonymous U2C authentication. However, at the time of the writing, the authors are not aware of any existing technology that would allow for correct anonymous U2C-preserving authentication. In this paper we instead assume that the users run the digital oblivion agent voluntarily and are not acting with malicious intent.
- Our system does not remove the data on the oblivion list from the real OSN. Therefore, an eavesdropper within the community could compare the real OSN and the oblivion view OSN, and localize content that should be forgotten through the differences. Also in this case we must rely on the good intentions of the participants. In general, until the data is completely removed from the underlying OSN (which in some cases can take a long time), it is difficult to completely protect against users with malicious intent. In this paper we help to reduce unintended exposure to content that users want forgotten.

Algorithm 8 Agent X receives disapproval of oblivion

```
Find all apparences of $H(F)$ on $L_O$ and consider the
    agents $B_1, \ldots, B_n$ whose tags resulted in the oblivion of
    $F$ and the agents $C_1, \ldots, C_m$ that previously disapproved
    the oblivion of $F$, all with identification registered at the
    entrance of $H(F)$ in $L_O$
if the trust combinations Trust($X, B_1, \ldots, B_n$) and
    TrustComb($X, C_1, \ldots, C_m, C$) evaluates in favour for dis-
    approving the oblivion of $F$ then
    Remove the entrance of $H(F)$ from $L_O$
else
    Add $Id_C$ to the list of agents who disapproves the
    oblivion of $F$
end if
Update the trust of the agents $B_1, \ldots, B_n$ and
$C_1, \ldots, C_m, C$
```

B. U2C Authentication Through Digital Signature and Watermarking

We claim that the correct and secure authentication of U2C R1 can be satisfactorily done by combining the cryptographic primitives perceptual hash, digital signatures and watermarking. Indeed, U2C authentication requires the user to sign the content in a way which makes it hard to remove the signature, which is exactly what is achieved when embedding the signature in the content using watermarking. We use perceptual hashes for the signature, since traditional hashes would be modified by the watermark, making them useless for our purpose.

The security of our implementation depends on the security of the involved primitives. Privacy issues can be avoided through the use of an anonymous digital signature scheme. Also the serial numbers should not be possible to link to the user, nor to other serial numbers generated by the same user.

C. U2C Authentication Through Tags and Trust Management

The authentication of U2C R2 can not be required to satisfy the same demands for security as does U2C R1. There is simply no way to securely ensure that some data refers to a specific individual. Consider for example a badly drawn caricature. It may require a big portion of cultural and ad-hoc reasoning to link the caricature to the individual it represents.

Security can only be ensured for specific well-defined properties. If we consider that a genuine tag of user U in an image F is a proof of a U2C R2 between U and F, then we can consider that security is achieved if it can be verified that the tag is genuine and that it links U and F. Since there are no security mechanisms implemented for tagging in current OSN, at the moment it is not possible to evaluate the genuine quality of a tag. It can be argued that security mechanisms could and should be added for tagging in future OSN.

Currently, the agents must rely on the trust they have in (i) the agent that requested oblivion, (ii) the agent of the user who tagged U, and (iii) the agents that disapprove the oblivion request. The design of the function that controls the combination of these trust values is therefore highly relevant for the security of the protocol.

D. Performance Analysis

To illustrate the feasibility of our system, we present a brief analysis of the resources consumed by the system. Table I shows the execution frequency of the different algorithms that constitute the system, as well as the resources that each of the algorithms consume. The execution frequencies are quantified as Seldom (less than once an hour), Common (several times an hour) and Frequent (several times a minute).

The statistics related to resource usage summarize the used building blocks and functions defined in Section III-A,

TABLE I. EXECUTION FREQUENCY AND RESOURCE CONSUMPTION OF EACH ALGORITHM.

Algorithm	Execution frequency	Executed algorithms (complexity)	Messages sent	Objects stored
1	Seldom: when a user joins the community	Keygen	-	1
2	Common: when the user uploads content	H, Sign, N, Embed	-	1
3	Seldom: when requesting oblivion of content	Steps 1-2: Recover Steps 3-9: H	- n_A	- 1
4	Seldom: when receiving oblivion request	Steps 1-10: H, Recover, Verify Steps 11-19: TrustComb	- -	- 1
5	Frequent: when viewing content on OSN	Recover, H, 2 searches (size o_A)	-	-
6	Seldom: when adding a friend	-	o_A (or $\lceil o_A/m \rceil$)	-
7	Seldom: when disapproving oblivion	H	n_A	1
8	Seldom: when receiving disapproval of oblivion	1 search (size o_A), 2 × TrustComb	-	1

each with their own complexity, the number of messages that must sent to other peers, and the overall addition in storage requirements caused by the algorithm. The presented values are quantified in terms of the resources consumed by one agent A, with n_A neighbors and o_A objects on its oblivion list. We also assume that m such object updates would fit in a message. Observe that apart from the resources listed here, the dealer also needs to maintain a list of public keys for the agents in the community. Note that our design ensures that the most frequent events (algorithms 2 and 5) does not result in any additional network traffic and only limited additional storage of information. Furthermore, messages are sent using limited broadcasting, such that the oblivion requests only are sent to friends (or alternatively some other set of users, depending on privacy settings, for example).

V. CONCLUSIONS AND FUTURE WORK

We have designed a system that provides digital oblivion for a community of users within an OSN. We have expressed digital oblivion for OSN in terms of authentication of user-to-content relations, and we have identified two user-to-content relations which we think are particularly important for digital oblivion: (U2C R1) having uploaded the content, and (U2C R2) presence of personal information in the content. We proposed several methods for authentication of these two U2C relations. For U2C R1, we proposed a combination of digital signatures and watermarking, employing perceptual hashes, and for U2C R2, we proposed facial recognition, semantics and tags, as indicators for personal information in content. We also described how trust can be used to manage security in authentication of U2C R2. In future work we will implement and evaluate a prototype of the system.

VI. ACKNOWLEDGMENTS

The authors would like to thank David Megías for useful suggestions. The first author is financed by the Swedish government through the National Graduate School in Computer Science (CUGS). Partial support by the Spanish MEC projects ARES (CONSOLIDER INGENIO 2010 CSD2007-00004) and ICWT (TIN2012-32757), is acknowledged.

REFERENCES

[1] Abril, D., Navarro-Arribas, G. and Torra, V. (2011) On the Declassification of Confidential Documents. Proc. MDAI 2011, Lecture Notes in Artificial Intellligence 6820, 235–246.

[2] Amanda Todd Suicide - FULL ORIGINAL VIDEO (12 October 2012) Available at http://www.youtube.com/watch?v=KRxfTyNa24A.

[3] Anandan, B., Clifton, C., Jiang, W., Murugesan, M., Pastrana-Camacho, P. and Si, L. (2012) t-Plausibility: Generalizing Words to Desensitize Text Transactions on Data Privacy, 5:3, 505–534.

[4] Chen, T., Wang, J. and Zhou, Y. (2001) Combined Digital Signature and Digital Watermark Scheme for Image Authentication, Int. Conferences on Info–tech and Info–net, Beijing.

[5] Cheng, Y., Park, J. and Sandhu, R. (2012) Relationship-based Access Control for Online Social Networks: Beyond User-to-User Relationships, Proceedings of the 4th IEEE International Conference on Information, Privacy, Security, Risk and Trust (PASSAT).

[6] Commission Proposal for a Regulation of the European Parliament and of the Council, art. 4(2), COM (2012) 11 final (Jan. 25, 2012), available at http://ec.europa.eu/justice/data-protection/document/review2012/com_2012_11_en.pdf.

[7] Domingo-Ferrer, J. Rational Enforcement of Digital Oblivion, Proceedings of the 2011 International Workshop on Privacy and Anonymity in Information Society (PAIS 2011), 2, 2011. ACM 2011.

[8] Druschel, P., Backes, M. and Tirtea, R. The right to be forgotten – between expectations and practice, Deliverable, ENISA, November 2012, available at http://www.enisa.europa.eu/activities/identity-and-trust/library/deliverables/the-right-to-be-forgotten.

[9] Hadmi, A., Puech, W., Said, B.A.E. and Ouahman, A.A. (2012). Perceptual Image Hashing, Watermarking - Volume 2, Dr. Mithun Das Gupta (Ed.), ISBN: $978-953-51-0619-7$, InTech, DOI: $10.5772/37435$. Available from: http://www.intechopen.com/books/watermarking-volume-2/perceptual-image-hashing.

[10] How does Facebook suggest tags? Facebook Help Center, Retrieved from https://www.facebook.com/help/122175507864081/.

[11] Jøsang, A., Ismail, R. and Boyd, C. (2007) A survey of trust and reputation systems for online service provision, Decis. Support Syst., 43:2, 618–644.

[12] Rosen, J. The Right to Be Forgotten, 64 Stan. L. Rev. Online 88, February 13, 2012.

[13] Ruohomaa, S., Kutvonen, L. and Koutrouli, E. (2007) Reputation Management Survey. The Second International Conference on Availability, Reliability and Security (ARES 2007), 103 –111.

[14] Strassberg, D.S., McKinnon, R.K., Sustaíta, M.A. and Rullo, J. (2012) Sexting by High School Students: An Exploratory and Descriptive Study, Archives of Sexual Behavior, 42(1), 15-21. DOI: $10.1007/s10508-012-9969-8$.

[15] Sweden students riot over Instagram sex insults page, BBC News, 18 December 2012. Available at http://www.bbc.co.uk/news/world-europe-20774640.

[16] Whittaker, Z. (4 October 2012). Facebook hits 1 billion active user milestone, CNET. Available at http://news.cnet.com/8301-1023_3-57525797-93/facebook-hits-1-billion-active-user-milestone/.

[17] Wolf, N. (26 October 2012) Amanda Todd's suicide and social media's sexualisation of youth culture, The Guardian. Available at http://www.guardian.co.uk/commentisfree/2012/oct/26/amanda-todd-suicide-social-media-sexualisation.