Can Messaging Applications Prevent Sexting Abuse? A Technology Analysis

Mirko Franco, *Student Member, IEEE,* Ombretta Gaggi, *Member, IEEE,* Claudio E. Palazzi, *Member, IEEE*

Abstract—The digital and mobile revolutions have changed the way people live their sexuality. Sexting, the practice of sending or receiving any sexually explicit content through mobile devices, has gained popularity, especially amongst teenagers and young adults, bringing several concerns, such as the uncontrolled spread of personal nude or semi-nude media without the owner's consent. Moreover, messaging applications generally used to communicate (and practice sexting) are not safe enough, e.g., they permit to send and forward any received content to anyone else. In this scenario, we believe that, beside education, technological solutions should be devised to avoid or limit sexting abuse. To this aim, we have developed *SafeSext*, a proof of concept messaging system, which also implements an image forwarding control feature. Through it, we have analyzed possible solutions, as well as their limits, in supporting a safer messaging environment where users retain some form of control over the forwarding of their self-generated sensitive contents.

Index Terms—messaging application, mobile, sexting

1 Introduction

According to the World Health Organization (WHO), sexuality is one of the essential aspects of being human. In the last decades, the growing spread of mobile devices and social media platforms have changed the way people communicate, search for new information, and create new relationships [1]. These revolutions have not spared sexuality, bringing several new opportunities and concerns. Indeed, mobile devices provide access to new ways of interacting with each other (even sexually) and to an almost endless storage of erotic material. In this scenario, *sexting* is a practice that has gained popularity especially, but not only, amongst teenagers and young adults and represents an interdisciplinary topic of interest involving psychology, medicine, sociology, and computer science.

We can define *sexting* as the practice of sending or receiving any sexually explicit content (e.g., text, images, audio, videos) through social media platforms, i.e., instant messaging systems, dating applications, social networks and so on [2]. Sexting allows to have sexual interaction without being physically close to each other, an important feature in particular situations, e.g., long-distance relationships, pandemic related lockdowns or people looking for sexual interactions with strangers without fully exposing themselves and avoiding the risks of any kind of contagion.

In general, sexting is a widespread phenomenon as revealed by a survey presented during the American Psychological Association's 123rd Annual Convention [3]. The authors asked to 870 heterosexual adults (aged between 18 and 82, average 32) to answer to a survey about sexting: 88% of the respondents had sexted at least once in their lives, with 82% having done it even during the past year.

Unfortunately, practitioners of sexting can become victims of the unwanted spread of their sensitive contents, when the receiver of these contents forwards them to others without consent. Clearly, the worst case is when images or videos are involved and this non-consensual pornography may happen after relationship breakups (a.k.a. revenge-porn), maliciously done by hackers, by mistake, or even selfishly done by receivers to boost their pride.

The relevance of the phenomenon is difficult to assess as it depends on whether the victims eventually become aware of it and whether they decide to report it to the authorities. Unfortunately, many factors can influence a victim towards not reporting this type of crime despite the suffered damage such as, for instance, the fear of further victimization and the embarrassment due to the private nature of the images.

Online studies conducted in Australia and in United States allowed to estimate the prevalence of non-consensual pornography; the results showed that about 10% had experienced the dissemination of their sexual images without having given consent [4], [5]. This is a very high percentage if we consider that non-consensual pornography has similar health consequences to sexual violence committed in person, such as dysfunctional behaviors (self-harm and alcohol abuse), anxiety, post traumatic stress disorder, depression and suicidal thoughts [6]. It can be a never-ending damage in which the victims constantly experience the fear of being recognized, not knowing who and how many may have seen their photos. Often, the victims feel a sense of powerlessness due to the impossibility of being certain to be no longer the object of dissemination. Clearly, the longer the sexual images have been out in the Internet without the victim knowing and acting upon it, the less it will be possible to block their spread.

For this reason, although education is the main weapon to prevent the risks of sexting and to fight against non-consensual pornography, we believe that technology should be investigated as well to determine whether it could be used to block the forwarding of sexting images or, at least, to promptly inform the owner of those images. Unfortunately, the current scenario is disheartening as commonly

- *M. Franco, O. Gaggi and C. E. Palazzi are with the Department of Mathematics, University of Padua, 35131 Padua, Italy (email: mifranco@math.unipd.it; gaggi@math.unipd.it; cpalazzi@math.unipd.it)*

Manuscript received April 19, 2005; revised August 26, 2015.

used messaging applications permit to send any content to anyone and, although some of them implement some features for a safer user experience (e.g., images that can be seen only once, i.e., Whatsapp, or expires after a predefined time interval, i.e., Snapchat), their safety with respect to non-consensual pornography is far from ideal.

Highlights and Main Contributions. We have developed *SafeSext*, a proof of concept of a messaging system contrasting non-consensual pornography thanks to a forwarding control algorithm. Our system recognises sexual images exploiting the Google Cloud Vision API and applies a perceptual hash function to associate an owner to each image. Then, it defines a forwarding policy that could block the forwarding attempt and/or alert the owner if the image is forwarded to someone else. To the best of our knowledge, this is the first attempt in this direction.

Structure of the Paper. The rest of this paper is organized as follow. Section 2 describes the scientific background of this work. A review of the safeness of the forwarding policies of some commonly used messaging applications is discussed in Section 3. Section 4 provides some background information about perceptual hashing functions. The general assumptions our work is based on are described in Section 5. Section 6 describes *SafeSext*, our forwarding control algorithm and some technical challenges along with possible solutions and their limitations. An evaluation of our system is reported in Section 7. Finally, we draw our conclusions and present some future directions in Section 8.

2 Related Works

Nowadays, teens spend a lot of time online and this habit has brought to light several concerns about their online safety. Some works try to understand the best approach to guarantee safety and security while adolescents use social media and Internet-enabled mobile devices. For instance, Wisniewski *et al.* [7] analyzed 42 features of 75 Android applications and mapped them against the Teen Online Safety Strategies (TOSS) framework. They considered mobile applications that have the purpose of promoting adolescents online safety and figured out that most of the considered applications favoured parental control over teen self-regulation. However, a teen-centric approach would be more effective in teaching teenagers the skills necessary to engage correctly and safely with others through mobile devices and online. In addition, these applications have a low adoptions rate. Indeed, Ghosh *et al.* [8] observed that teenagers and children often dislike these applications, especially those adopting the parental control, because they find them too restrictive and too invasive for their privacy. Some works tried to co-design such applications with teenagers and children to find a balance between their need for privacy and their safety [9], [10].

The relationship between the Internet, social media and sexuality has gained the interest of the Human-Computer Interaction research community. Indeed, teens live part of their sexual experiences through the Internet, raising several concerns about their safety. For instance, sexting can have serious consequences, such as the spread of personal contents without the owner's consent, which may lead to psychological issues, (cyber)bullying, self-harm behaviors, or even suicide [11], [12]. On the other hand, online sexual experiences have become an important part of teens' lives. Hence, we need to rethink social media used everyday to face the new challenges posed by this particular use case [13]. Hartikainen *et al.* [14] focused on some messaging applications describing the design implications that emerged from their analysis of comments in a teen peer mental health support forum. According to their findings, messaging applications should allow to easily obscure faces in images containing nudity, as well as support their users in case of sexting abuse and advise them on how to behave correctly online. Particularly noteworthy is the recommendation of involving teenagers and sexual health experts in the design process of such applications.

Some researchers proposed computational approaches to detect nudity and/or skin. For instance, Wang *et al.* [15] developed a nude image recognizing algorithm based on the navel and some high-level hardcoded body features, which is able to detect nudity only if a body is completely nude. Santos *et al.* [16] presented a solution that allows recognizing exposed private parts of one's body even when mostly covered, mixing high- and low-level features. Instead, Sevimli *et al.* [17] described an approach able to classify images in five different classes (normal, swimming suit, topless, nude, and sexual activity), employing four descriptors in their algorithm. A comprehensive survey on algorithmic and computational approaches to detect nudity and skin that can be used to prevent sexting adolescent behaviors is presented in [18]. Razi *et al.* [19] reviewed 73 papers on computational approaches for online sexual risk detection, employing a human-centered lens.

Some works proposed serious games designed to tackle sexuality-related topics with teenagers. For instance, Wood *et al.* [20] presented a multiplayer mobile game to encourage talks about sex and sexuality in a group of teens, acknowledging that a permissive approach is more effective to improve teenagers' sexual health than a restrictive one. Instead, Guava *et al.* [21] developed a serious game to educate players about contraception and sexually transmitted diseases. However, education has poor results in a short time. Hence, teens need systems to safely explore their sexuality. In [22], we proposed some guidelines for developers to design safe messaging applications that allow for some form of control in the forwarding of media.

Safety issues are particularly important for people with disabilities that could be stakeholders of the proposed system [22]. Indeed, although little is known about the relationship between sexting and disability, it is not true that they do not have sexual experiences, fantasies and expectations like people without impairments often think [23]. In this scenario, sexting can be helpful for people with disabilities, especially physical ones, to gain confidence with their bodies and as a first step toward sexual experiences in real life. Moreover, through sexting, they can show only what they want of their body, which is not possible in real life [24].

In this context, we proposed some guidelines to build messaging systems safer by design for sexting in [22]. In the current work, we analyze possible solutions as well as their limits, thus opening new research directions, in creating a messaging environment where users retain control over the forwarding of their self-generated sensitive contents. Consequently, another important contribution is the demonstration that problems related to non-consensual pornography are not just a matter of how people use a (supposedly) neutral-by-nature tool but, rather, that technological solutions can be employed to improve the safety of popular applications and, thereby, at the time of writing, the inertia of social platforms can be considered co-responsible for possible sexting-related abuse harming users.

To the best of our knowledge, this is the first attempt in this direction. Therefore, our work advances the current state of the art, proposing a proof-of-concept messaging system, which provides technological support to contrast the forwarding of sexting-related images. Through our study, we discuss technical challenges and possible solutions, also analyzing their effectiveness/limits and opening new research directions for scientists and social media creators.

3 Analysis of Existing Applications

Many popular mobile applications contain some messaging functionality. Some of them are introducing features to provide a safer user experience. We have hence analyzed the messaging functionalities of some commonly used social media applications from the standpoint of sexting safety. In particular, we considered the following features:

- if the application is end-to-end encrypted or not;
- the possibility to delete messages already sent for both the parts of a conversation;
- if it is possible to set an auto deletion timer for messages;
- if the application notifies the users when someone takes a screenshot or starts a screen recording;
- whether it is possible to forward content or not.

TABLE 1
Summary of the Features of the Analyzed Applications

Name	E2E Encr.	Message Deletion	Auto Deletion Timer	Screenshot Alert	Fwd
Badoo	No[a]	No	No	No	Yes
Instagram	No	Yes	Yes[b]	Yes[e]	Yes[f]
Snapchat	No	Yes	Yes	Yes	Yes
Telegram	Yes[c]	Yes	Yes[c]	Yes	Yes[f]
Tinder	No[a]	No	No	No	No
Tumblr	No[a]	No	No	No	Yes
Whatsapp	Yes	Yes[d]	Yes[b]	No	Yes

[a] not available [b] only for photos and videos [c] only for secret chats [d] only within 7 minutes from message generation [e] only for time-limited media [f] unless for time-limited media

Since pure messaging applications (e.g., Whatsapp, Telegram) are not the only ones that contain messaging features, we also selected dating and social networks apps: in alphabetic order we analysed Badoo, Instagram, Snapchat, Telegram, Tinder, Tumblr, and Whastapp as representative of the vast plethora of possible choices.

Although all these applications provide some messaging functionalities, there are several differences in their features and the underlying philosophies: e.g., Tinder does not allow to send media (i.e., images, videos, etc.) at all, while all the others platforms permit to send different kinds of media.

Several applications have introduced end-to-end encryption, making the contents readable only to the sender and the receiver: e.g., Whatsapp is end-to-end encrypted, whilst Instagram has planned to introduce this additional guarantee for privacy and security in 2023. Telegram provides encryption only for secret chats. On the other hand, Tumblr serves all over HTTPS by default. Hence, all the content is encrypted only between client and server, without any further level of security. Many applications lack transparency on their policies, especially for privacy and encryption, making it difficult for users to acquire information [25] [26]: Badoo, Snapchat, and Tinder do not provide this information in the official description.

Among the considered platforms, Instagram and Telegram let users delete already sent messages for both sides of a conversation, providing a helpful tool in case of regret. This feature is not provided by Badoo, Tinder, and Tumblr, whereas Whatsapp permits such operation within 7 minutes from the message generation. Snapchat adopts a more complex behavior: the application automatically deletes all the messages already read by the receiver when the user closes the chat. However, users can save messages and, in this case, the deletion requires explicit action, such as to tap on a dedicated button to unsave (and hence delete) messages.

This functionality for non-saved messages is a sort of auto-deletion timer. Some other platforms provide similar features. For instance, on Instagram, it is possible to send media (images and videos) that can be viewed only once or twice. Similarly, Whastapp recently introduced the same feature but limited the number of views to one. On the contrary, users can set an auto-deletion timer for any message on Telegram secret chats.

Another important issue is the control against screenshots or recording of a conversation since such actions can be a threat when dealing with personal and sensitive content. Snapchat is the only application that shows two different alerts for screen recording and screenshot actions so users can be aware of what is happening. Instagram notifies users only when time-limited media are involved, whilst on Telegram, such functionality is available only for secret chats. Table 1 shows a summary of the analyzed functionalities of the considered applications.

Except Tinder, all other considered applications permit forwarding[1] any content to anyone, providing an easy way for a malicious user to disseminate personal content (e.g., self-generated nudes). Indeed, none of such applications implements some form of control on the forwarding of media. Instead, our system proposes a step forward in this direction, advancing the current state of the art as well as opening new research directions.

4 Background on Perceptual Hashing

Our *SafeSext* system needs a computationally efficient way to represent and compare images. Our platform uses perceptual hashing to detect not only bit-level identical image copies but also edited ones.

[1] By forwarding, we intend either the presence of a forwarding feature or the possibility to save and send content later (e.g., images, videos, etc.).

4.1 Theoretical Foundations

In recent years, the rapid development of capable multimedia technology and the popularization of image manipulation tools (e.g., Photoshop) have made it possible to easily modify images, even considering the capabilities of current mobile devices. For instance, the scenario where a malicious user edits a personal photo of someone else leads to dangerous consequences. Therefore, in many applications, including our *SafeSext*, the verification of the authenticity of images has become a relevant issue.

A hashing function is a one-way mapping that can transform an image (in general some data) into a short sequence of bits of fixed length (image hash that is also known as image fingerprinting). Perceptual hashing functions have to fulfil the following requirements (where P denotes probability, X, $\hat{X}$, and Y are images, α and β are hash values, and $\{0,1\}^L$ represents binary strings of length L) [27]:

1) Equal distribution of hash values, $\forall \alpha \in \{0,1\}^L$

$$P[H(X) = \alpha] \approx \frac{1}{2^L} \qquad (1)$$

2) Pairwise independence of visually different images X and Y, $\forall \alpha, \beta \in \{0,1\}^L$

$$P[H(X) = \alpha | H(Y) = \beta] \approx P[H(X) = \alpha] \qquad (2)$$

3) The distinction of visually different images X and Y

$$P[H(X) = H(Y)] \approx 0 \qquad (3)$$

4) Invariance for visually similar images X and $\hat{X}$

$$P[H(X) = H(\hat{X})] \approx 1 \qquad (4)$$

The last two requirements are the fundamental properties of perceptual hash functions. In other words, an image perceptual hashing function maps visually similar images to similar hash values and visually different images to different fingerprints. Hence, not only do we identify bit-level identical photos, but we also detect visually similar ones. As a consequence, by using a perceptual hashing function, our system becomes more effective and safer.

Over the years, scientists have proposed several functions for perceptual hashing that address different applications and transformations of the images. The most representative examples are the Block Mean Hash [28] and the Color Moment Hash [29], both implemented by the OpenCV framework, one of the most important in computer vision. Such type of function, hereafter referred to as a conventional hashing function, requires designing and extracting the features manually, hence not representing the image content properly. Consequently, traditional hashing schemes cannot obtain the optimal trade-off between robustness against image manipulations techniques and discrimination capabilities, whilst both properties are fundamental when facing the challenge of image authentication. Furthermore, Monga *et al.* [30] have proven that the problem of simultaneously optimizing a features extractor and a hash generator separately designed is NP-complete.

Thanks to the explosion of the number of new approaches based on deep learning and the excellent representation capabilities of neural networks, several new perceptual hashing schemes have been proposed [31]. Some representative examples are based on Convolutional Neural Networks (CNNs) [32] or on autoencoder and feedforward neural networks [33]. This kind of approach represents the state-of-the-art solution in reaching remarkable results in both robustness and discrimination capabilities; we hence decided to employ it at the core of our system.

4.2 The Fingerprinting Function of SafeSext

Since *SafeSext* needs a hashing function with excellent performance to be effective and have good scaling capabilities, we have chosen a function based on neural networks. In particular, we have adopted the state-of-the-art model proposed in [33], which was characteristics useful for our system (e.g., good robustness and discrimination capabilities, few false positives, etc.). Moreover, the code is available, we need to port it from MATLAB to Python using the framework PyTorch, which allows us to use the backpropagation algorithm, the optimizers, and the loss functions as features available out of the box. Furthermore, this choice is in line with the deep learning research community.

Other differences regard the training procedure, which is composed of a pretraining step and a fine tuning one as described in [33]. Yet, we have modified the learning rate of the fine tuning from the original proposal of $2 * 10^{-3}$ to $5 * 10^{-5}$ since we empirically observed better performance with our set of image manipulations (JPEG Compression, Scaling, Flipping, Rotation, Brightness, Contrast, Circle Mark). The data used for the training procedure and for testing the model are a subset of the COCO dataset [34]. We selected transformations generally and easily available to user employing classic image editing app on their mobile devices. Yet, the algorithms for a broad part of the possible modifications are either proprietary or without a commonly accepted definition. For instance, this is the case of brightness, which could have a relevant impact on the performance of the hashing function in a real scenario and deserves further investigation in the future.

Before presenting the results achieved by the model, we define the concept of distance between two hash values. Given two image hash values h_1 and h_2, their distance indicates how similar the corresponding images are and it is computed as:

$$D(h_1, h_2) = \sum_{i=0}^{N} (h_1^{(i)} - h_2^{(i)})^2 \qquad (5)$$

where N denotes the length of the hash values, while $h_1^{(i)}$ and $h_2^{(i)}$ are the i-th elements of h_1 and h_2, respectively. If the distance D between the two hash values is smaller than a prefixed threshold, the two images can be considered perceptually identical, otherwise they are perceptually distinct.

The main difference with respect to the original work in [33] is the transformation of the images used to train the network. Ideally, we would want a model robust against all possible modifications; unfortunately, this is not possible. Regarding our scenario, we have restricted our expectations in training a secure model against manipulations commonly available on mobile devices, as it is easier for users to access them. Table 2 shows the details of the transformations used to train and test the neural network of our system.

TABLE 2
Details of Content-Preserving Image Operations and Minimum, Maximum, Mean and Standard Deviation of the Distances between the Original Image and the Result of Every Considered Manipulation

Manipulation	Strength	Min	Max	Mean	Std. Dev.
JPEG Compression	Quality Factor $\in \{1, 5, 10, 30, 50, 70, 90, 100\}$	$2.51 * 10^{-7}$	$7.07 * 10^{-1}$	$2.07 * 10^{-2}$	$6.40 * 10^{-2}$
Scaling	Ratio $\in \{0.2, 0.4, 0.5, 2, 4\}$	$1.39 * 10^{-3}$	$8.01 * 10^{-1}$	$4.16 * 10^{-2}$	$8.00 * 10^{-2}$
Flipping	All the possibilities	$5.71 * 10^{-2}$	9.60	2.18	1.95
Rotation	Angle $\in \{0, 5, 15, 30, 45, 90, 135, 180, 225, 270, 315\}$	0.0	20.06	3.53	3.27
Brightness	Offset $\in \{-80, -60, -40, -20, -10, 10, 20, 40, 60, 80\}$	0.0	3.11	$1.15 * 10^{-1}$	$2.68 * 10^{-1}$
Contrast	Offset $\in \{-80, -60, -40, -20, -10, 10, 20, 40, 60, 80\}$	$1.17 * 10^{-6}$	2.22	$3.12 * 10^{-2}$	$1.10 * 10^{-1}$
Circle Mark	Thickness $\in \{5, 10\}$	$8.69 * 10^{-3}$	3.53	$3.67 * 10^{-1}$	$4.70 * 10^{-1}$

(a) Airplane (b) Baboon (c) Boat (d) House (e) Peppers

Fig. 1. Standard test images.

To perform our test campaign, we adopted a method generally employed when considering image authentication. We computed the distances between the five original images shown in Fig. 1 and their perceptually identical versions obtained through the aforementioned image manipulation. Table 2 details the employed manipulation parameters and distance statistics with respect to the original image. We report the most representative results in Fig. 2, Fig. 3, and Fig. 4, where each chart shows the distances between the hash values of the original images and the hash values of the same photos edited with the specified transformation, using the values specified in the x-axis. Generally, the shorter the distances are (i.e., the closer their values are to 0), the better the perceptual hashing function behaves. For instance, we can see in Fig. 2 (*scaling*) that distances are close to 0, meaning that the hash values of the original image and its corresponding manipulated version are almost the same. The same situation happens for *JPEG compression*. Instead, our perceptual hashing function is not so robust against *rotation* since the distances shown in Fig. 3 are not close to 0.

To assess the robustness of our hashing function, we randomly selected 100 images from the COCO dataset and computed minimum and maximum, average and the standard deviation of the distances between the original image and its modified version. We report the results in Table 2, which shows that the average distance for almost all the manipulations is below 0.40, meaning that our perceptual hashing function has good robustness against the considered content-preserving manipulations.

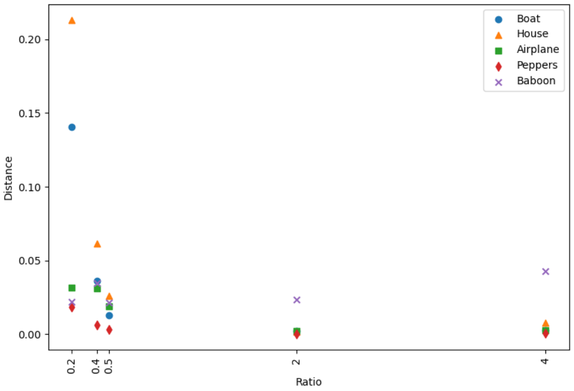


Fig. 2. Distances for scaling.

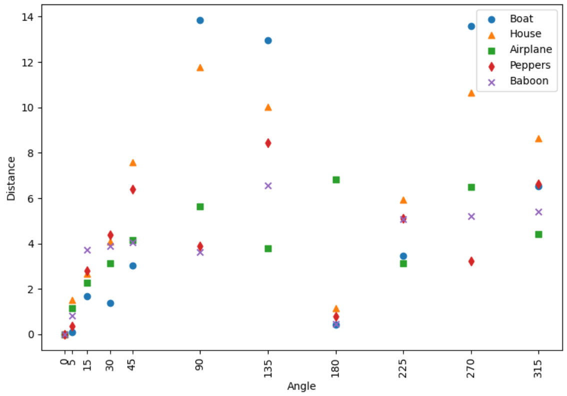


Fig. 3. Distances for rotation.

To evaluate the performance of our fingerprinting function, we use a fixed set of thresholds τ_i, which embody the possible distances between the hash values used to classify whether images are similar or not. In our case, we employed 1000 values between the minimum and the maximum distances obtained with our test set. We compute the F_1 score for each classifier (i.e., considering each possible τ_i as threshold) as:

$$\frac{2 * [1 - FAR(\tau_i)] * [1 - FRR(\tau_i)]}{[1 - FAR(\tau_i)] + [1 - FRR(\tau_i)]} \tag{6}$$

Then, we select the classifier with the best score. In other words, we choose the threshold that maximizes the F_1 score. The model selected by this procedure has an F_1 score equal to 0.89 and suggests 2.75 as the threshold. It has a *FAR* of $6.98 * 10^{-2}$ and a *FRR* of $1.37 * 10^{-1}$. These values show that the model performance is acceptable. However, our system needs to have strong scaling capabilities, and these results could not be enough. We have also tried to use the same parameters for the neural network as the ones

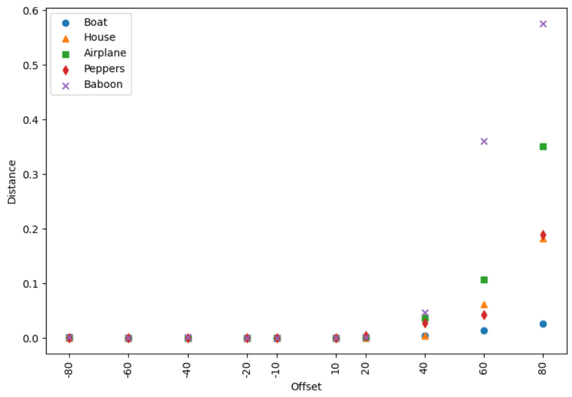


Fig. 4. Distances for contrast.

used in [33] but they generate a higher *FAR* and a less stable course of the loss, error and F_1 score curves during training and validation. Indeed, the complexity of the image manipulations we consider requires proper settings of the parameters as the one we used to improve the performance of the system. Therefore, the neural network at the basis of our fingerprinting function does not reach the results shown in [33] and we suppose that the reason is that we use more complex manipulations than the ones used in the original paper. However, a formal proof of this assumption is out of the scope of this work.

5 Assumptions

Sexting-related issues are mainly due to unauthorized forwarding of personal content. Forwarding can happen either thanks to an ad-hoc functionality of a messaging application or to the possibility of saving content on the device and sending it later. Thus, a messaging platform that does not offer these features would clearly help solving the problems related to sexting. Unfortunately, it would also not be possible to forward content which is not sensitive (e.g., a sunset, a landscape, etc.), thus jeopardizing the experience offered to the users by the messaging application. Similar, even just the absence of the possibility to save content on the device would simplify the design of a safer system. However, in this work, we decided to consider the most general case, which is the one where users can save the content on their device (e.g., in the gallery application). In this way, we aim to propose a solution that addresses difficult scenarios, avoiding unrealistic simplifications, while minimizing the negative effects on user experience.

6 Description of SafeSext

We designed and developed a proof of concept of a messaging system (and an Android application) attempting to be safer by design for sexting thanks to a forwarding control feature. Its goal is to reduce and prevent the spreading of photos containing self-generated nudity without the consent of their owners. We hence investigated technological solutions that could provide protection against sexting abuse in the form of non-consensual pornography.

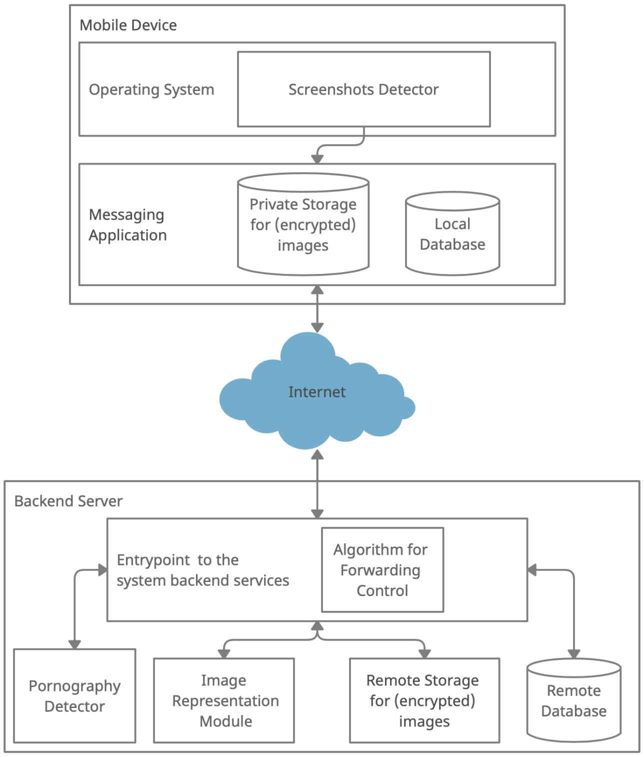

Mobile Device
Operating System
Screenshots Detector
Messaging Application
Private Storage for (encrypted) images
Local Database
Internet
Backend Server
Entrypoint to the system backend services
Algorithm for Forwarding Control
Pornography Detector
Image Representation Module
Remote Storage for (encrypted) images
Remote Database

Fig. 5. Architecture of SafeSext.

Besides the forwarding control functionality, *SafeSext* also includes other features to support users during their experience with our platform. For instance, our system lets them delete messages in a conversation for both sender and receiver, controls and reports screenshots activity (e.g., log of the actions, screenshot actions disabled), and supports users in case of issues related to sexting.

Our system includes a mobile application that allows the exchange of messages between users by communicating with a server through the Internet. Components of the mobile application and of the server are shown in Fig. 5.

The mobile application includes a local database and a private storage. *SafeSext* saves messages and contacts in the former, while the latter is employed to store images. Moreover, it provides some form of control on the screenshot actions. For instance, thanks to a notification triggered by the operating system, it is possible to know when users take a screenshot. However, such an operation could not be always possible: e.g., our specific Android implementation does not allow to take screenshots, but the same feature is currently not implementable in iOS and requires further investigation or some policy change by Apple.

Mobile application interacts with our backend server by calling the right HTTPS endpoint, a sort of entry point to all the services of our system. For instance, the API permits registering users into the system, sending messages, requesting the deletion of content, etc. The most important features of the backend server are:

a) the forwarding control algorithm;
b) the forwarding policies;

c) the identification of images whose sharing have to be restricted;
d) the image representation module, which implements a perceptual hashing function to compute the representation of the images.

The remote database contains information about users of the system and hash values of the images and their owner. The forwarding control algorithm uses the hash values to check the owner of an image and apply the forwarding policies. The system saves the photos in remote storage thus permitting the update of the fingerprinting module in the future, as described in Section 6.4. Yet, we can anticipate here that other design choices are possible, even considering the possible privacy drawbacks of this part of the system. For instance, users can be asked to upload images on the fly when the fingerprinting function has to be updated.

Our system, and in particular our forwarding algorithm, is completely transparent for the users until a suspicious action happens. Indeed, no preliminary photo upload or user action is required.

6.1 Algorithm for Forwarding Control

Our system imposes restrictions on the forwarding of images thanks to the algorithm for forwarding control, which is the core of *SafeSext*. In this section, we describe it in detail.

When a user sends an image, the algorithm checks whether such an image is relevant for the system or not. Section 6.3 discusses what this means for a photo in *SafeSext*, but we can anticipate that the algorithm considers relevant a picture when it contains nudity of any sort. If the image is not relevant, the system sends it to its receiver since it is not sexual-related. Otherwise, the image representation module computes the hash value of the considered photo and the hash values already known by the system are retrieved.

The system calculates the distances between the new hash value and the ones previously retrieved[2]. If an image with a distance below a predefined fixed threshold exists, the system controls whether the sender or the receiver of the image is its owner. In such a case, the algorithm sends the photo as shown in Fig. 6. Otherwise, if the distance is below a predefined threshold, and the owner of the considered photo is neither the sender nor the receiver, the system could adopt different behaviors according to the implemented forwarding policies. For instance, one approach could specify to not proceed with the sending, while another policy could impose to notify the owner of the considered image but send it anyway as discussed in the next section. If none of the distances between the hash value of the considered image and the ones known by the system is below the fixed threshold, the system saves the hash value of the currently considered picture and sets the sender as its owner. Then, the image is sent out to the receiver.

Although the evaluation of the relevancy of an image may be wrong, an error in the decision on whether two hash values correspond to the same image or not is way more disrupting. In case of a false positive, the system executes the predefined forwarding policy even though not necessary. Instead, in case of a false negative, the system inserts the hash value of the image along with its owner, thus leading to the presence of two owners for the same photo. For these reasons, as discussed in Section 4.2, we need to lower as much as possible the error rate of the employed fingerprinting function.

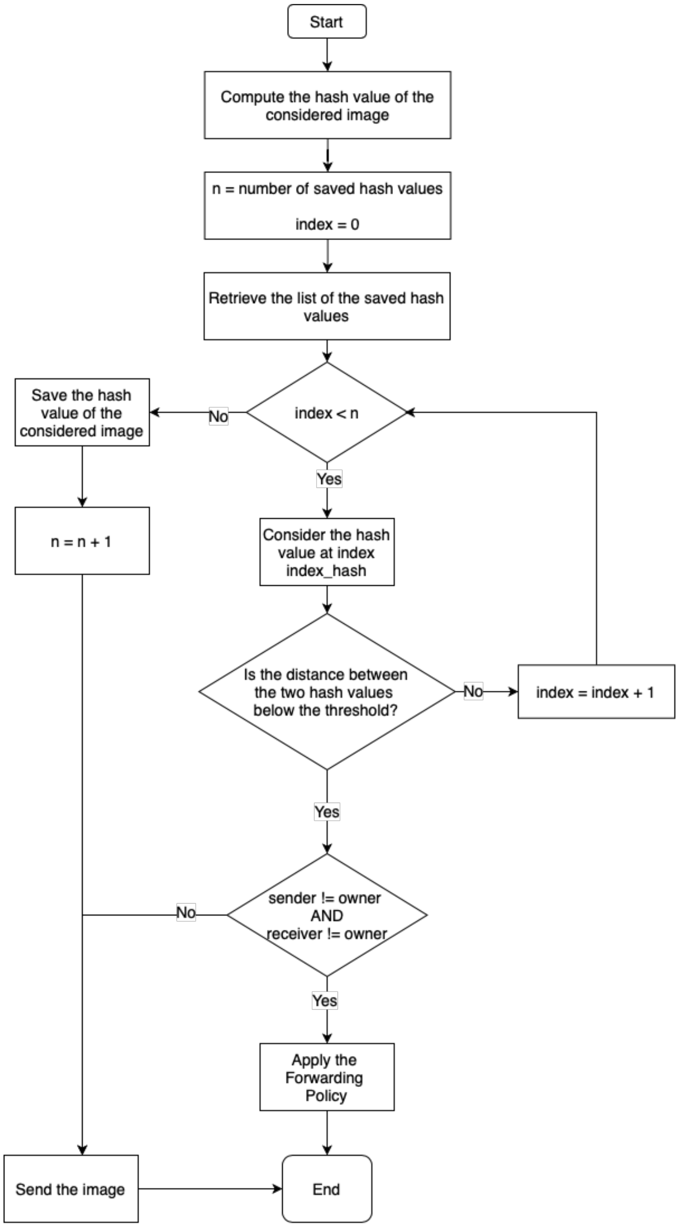

Start
Compute the hash value of the considered image
n = number of saved hash values
index = 0
Retrieve the list of the saved hash values
index < n
No
Save the hash value of the considered image
Yes
n = n + 1
Consider the hash value at index index_hash
Is the distance between the two hash values below the threshold?
No
index = index + 1
Yes
sender != owner AND receiver != owner
No
Yes
Apply the Forwarding Policy
Send the image
End

Fig. 6. Core of the forwarding control.

We must note here that this approach can be generalized to several other kinds of media, applications, and scenarios. For instance, it could be interesting to add the forwarding control of videos in *SafeSext*, as videos are actually frequently used for sexting. The algorithm does not need to change significantly; just some of the steps need to be adapted to the video scenario. In particular, it is necessary to define when two media have to be considered as similar and specify a function to represent them, along with a proper distance measure.

[2] The only case in which the system does not retrieve any hash value is when the first user sends the first image, so we can assume that the set of retrieved hash values is not empty.

6.1.1 Computational Overhead Assessment

The forwarding control algorithm is a very central part of our system. Its execution time directly impacts the user experience and hence the spread of our proposal. Therefore, both theoretical and empirical analysis of its time complexity are necessary. As can be easily understood from Fig. 6, our forwarding control algorithm has time complexity O(n) in the worst case, which happens when the considered image is not already known. Indeed, the algorithm iterates over all the known hash values in that case. Yet, this time complexity can be prohibitive in case a large number of hash values is saved into the system, especially considering the interactive nature of our application. We can hence adopt other searching strategies (e.g., binary search, O(lg n)) and/or appropriate data structures.

We now report the results of some experiments where we calculated the computation time of various parts of our algorithm. The first step of the forwarding algorithm is deciding whether an image is relevant or not by employing the Google Cloud Vision API. We have hence computed the maximum, minimum, mean, and standard deviation of the response time (i.e., the computation time plus the network delay) of such an API, applying it to 1000 images at different times during the day. Table 3 presents the results. Furthermore, if an image is considered sensitive, the system computes its hash value using our fingerprinting function. Therefore, we have calculated the computation time of the hash value generation, including the preprocessing time of the image, considering the same 1000 photos used in the evaluation of the previous step. Our experimental results show that the time needed for this step ranges from 4.67 * 10^{-3}s to 2.30 * 10^{-1}s, yet it is 9.95 * 10^{-3}s on average and presents a standard deviation equal to 8.60 * 10^{-3}s. Finally, we have computed the time necessary to search amongst the known hash values, considering the worst-case, which should be the most frequent one (i.e., when the search fails). To this aim, we have considered both a linear search in the list of known fingerprints and a dichotomic search. In the latter case, if the search fails, the insertion has time complexity O(n) in the worst case since the list is ordered (with respect to O(1) amortized in the former case). Yet, the insertion may be deferred and executed in the background. Thus, as demonstrated by our experimental results, the binary search increases the performance of the algorithm. We report our numerical results in Table 4 and Table 5. The mean of the computation times of the linear search increases by a factor of 10 every time the input increases by the same factor, in accordance with its theoretical time complexity. Instead, considering the binary search, the mean of the computation times maintains the same magnitude order for all the considered input lengths, thus confirming our hypothesis and theoretical analysis.

We must note here that the search can be parallelized, thus lowering the its computation time. Furthermore, the system can employ some heuristics in the search (e.g., consider the contacts of the sender before other users).

TABLE 3
Image Relevancy Evaluation: Response Time

Time	Min (s)	Max (s)	Mean (s)	Std (s)
9 A.M.	2.02 * 10^{-1}	8.21	3.31 * 10^{-1}	2.59 * 10^{-1}
6 P.M.	2.48 * 10^{-1}	1.32	4.00 * 10^{-1}	8.80 * 10^{-2}
9 P.M.	2.43 * 10^{-1}	2.43	4.04 * 10^{-1}	1.21 * 10^{-1}

TABLE 4
Linear Search: Computation Time

Values	Min (s)	Max (s)	Mean (s)	Std (s)
1000	1.74 * 10^{-2}	1.00 * 10^{-1}	1.91 * 10^{-2}	3.33 * 10^{-3}
10000	1.71 * 10^{-1}	5.71 * 10^{-1}	1.92 * 10^{-1}	3.33 * 10^{-2}
100000	1.54 * 10^{-1}	4.33	2.22	4.39 * 10^{-1}

TABLE 5
Binary Search: Computation Time

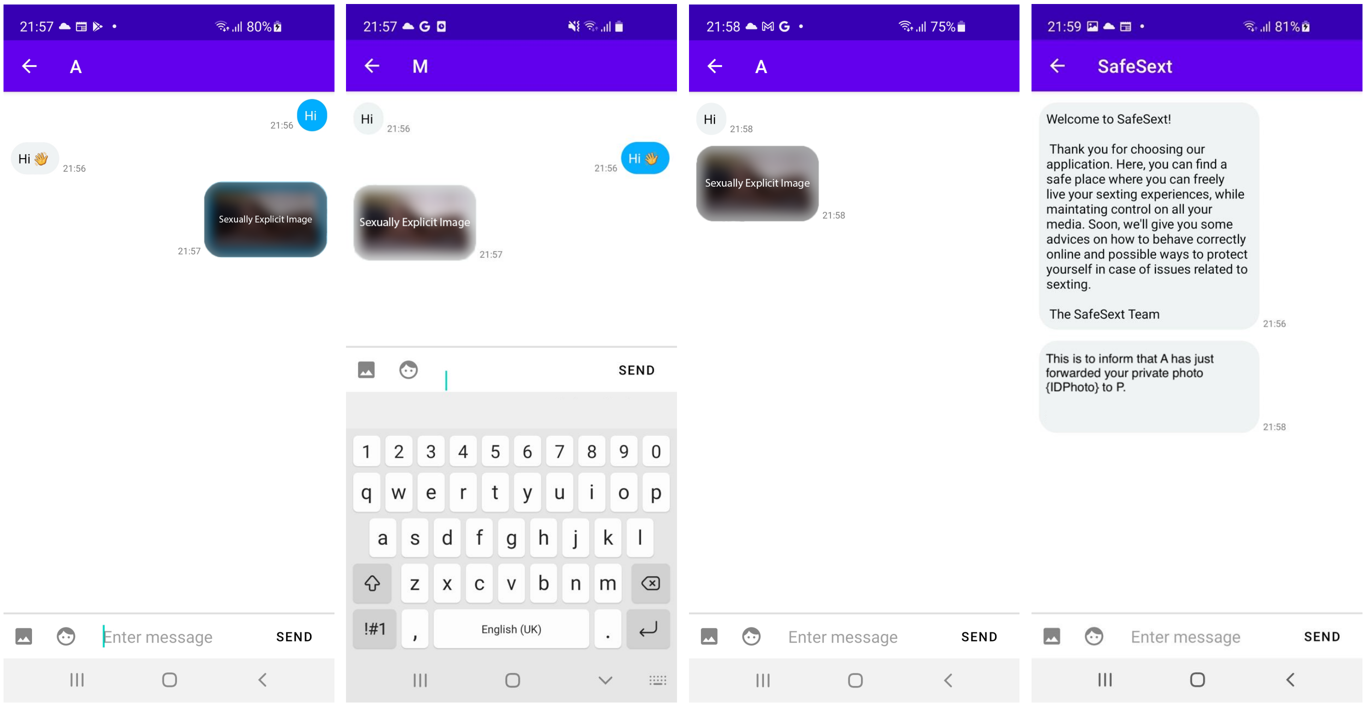
Values	Min (s)	Max (s)	Mean (s)	Std (s)
1000	2.06 * 10^{-3}	5.41 * 10^{-2}	2.62 * 10^{-2}	4.04 * 10^{-3}
10000	1.76 * 10^{-3}	1.33 * 10^{-1}	4.05 * 10^{-2}	6.83 * 10^{-3}
100000	3.15 * 10^{-3}	1.06 * 10^{-1}	5.08 * 10^{-2}	8.77 * 10^{-3}
1000000	2.32 * 10^{-3}	3.48	5.17 * 10^{-2}	1.10 * 10^{-1}
10000000	5.20 * 10^{-3}	3.36	7.50 * 10^{-2}	1.98 * 10^{-1}

6.2 Forwarding Policies

Our system can adopt different policies to contrast the attempts of forwarding a sexting-related image without being the owner. The optimal behavior can depend on several factors, such as the performance of the fingerprinting function. Therefore, we have included in our system two possible forwarding policies: notification and blocking. Probably, the combination of the two is the best option in an ideal system, although other ones might be proposed even in combination.

The first policy, if adopted alone, allows the forwarding of any image but notifies its owner when such a photo is a personal/sensitive one and someone who is not the owner is sending it out. In Fig. 7, we show a sequence of events that triggers a notification. Basically, M sends a self-generated nude image to user A (Fig. 7a), who receives it (Fig. 7b) and forwards it to another user (Fig. 7c). As a consequence, the system notifies user M (Fig. 7d).

Such a policy seems not to be fully coherent with the purpose of our system; yet, some users might prefer it or, as said, the two policies could be used in combination. Employing solely forwarding notifications, without blocking the forwarding of an image, can avoid annoying blocks in case of false positives. Indeed, although *FAR* and *FRR* reported in Section 4 are low, about four photos over one hundred will be considered as false positives. This number has to be considered in the context of a messaging applications where users tend to send and receive hundreds of images every year. Thereby, if we do not leave to the users the possibility to configure the system to only employ forwarding notifications without blocking, these false positives (if felt annoying) may lead users to uninstall *SafeSext* in a period of their life when not thinking of sexting (and its risks) at all and switch to a different platform. As users tend to not continuously switch amongst messaging applications, the latter will then be used when/if users will engage in sexting, leaving them without any protection (not even notifications) against unwanted forwarding. Instead, as already mentioned in Section 1, it is crucial for the victims of non-



Welcome to SafeSext!

Thank you for choosing our application. Here, you can find a safe place where you can freely live your sexting experiences, while maintaing control on all your media. Soon, we'll give you some advices on how to behave correctly online and possible ways to protect yourself in case of issues related to sexting.

The SafeSext Team

21:56

This is to inform that A has just forwarded your private photo {IDPhoto} to P.

21:58

(a) M sends a self-generated nude image to A.

(b) A receives the image from M.

(c) Someone else receives the M's image forwarded by A.

(d) M receives a notification because A just sent the image to someone else.

Fig. 7. Example of notification after a forwarding.

consensual pornography to become immediately aware of the crime so as to act immediately and block the spread of the images before they get replicated so many times over the Internet to render any counteraction ineffective.

The second proposed policy simply blocks the forwarding of a relevant image (e.g., a self-generated nude), except when the sender is its owner. Fig. 8 shows an example of implementing such a policy. In particular, user M sends a self-generated nude image to A (Fig. 8a), who receives it (Fig. 8b) and tries to forward it to someone else. However, the operation is forbidden and the system shows an alert to the forwarder (Fig. 8c).

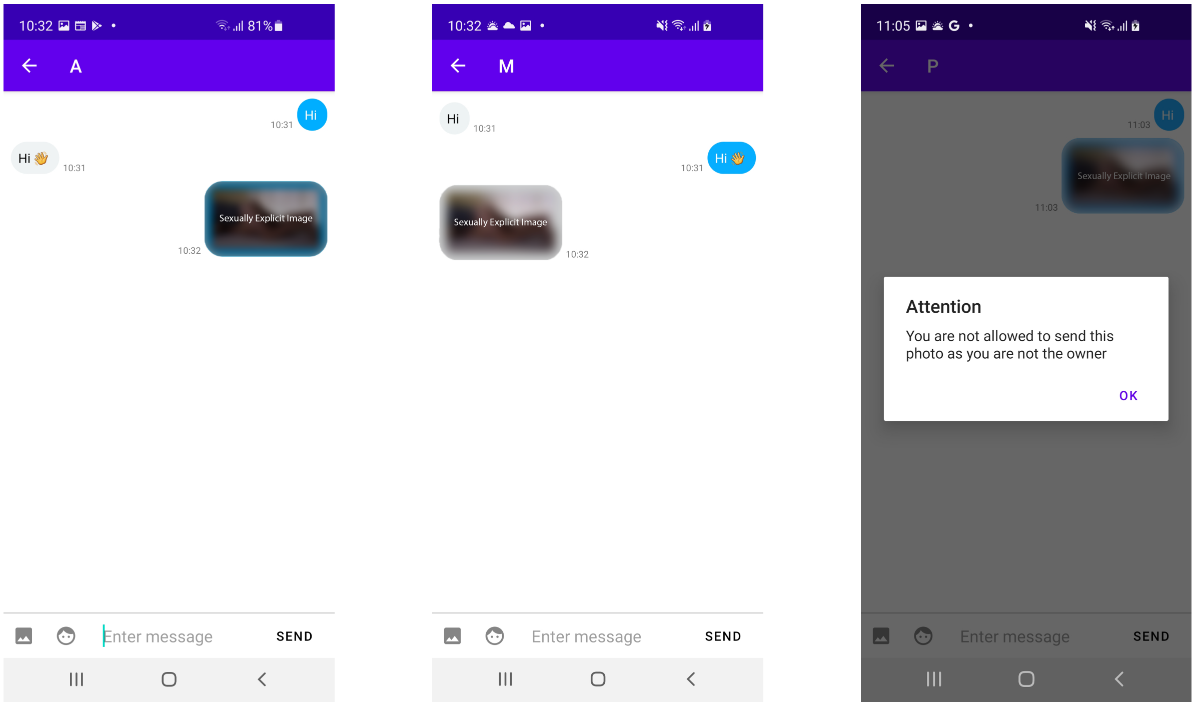
The blocking forwarding policy and notifications can also be used together to make the owner of the image aware of the attempt and act accordingly: granting forwarding permission (e.g., if the image is actually a false positive or the owner agrees with the forwarding), contact the police, etc. Other policies are possible as well. For instance, people who share images depicting their faces are more at risk in case of revenge porn as it is easier to identify them; this is also true for people with identifiable marks (e.g., tattoos). In a scenario where there is no identifiable information in the sexual picture, the system can adopt a different forwarding policy. However, we must note that an image can generally reveal various types of sensitive information and people are unaware of how images can compromise privacy [35], [36]. Therefore, other countermeasures should be adopted to protect users besides a forwarding policy. A comprehensive survey on image privacy in online social network is presented in [37].

Some further considerations are needed about notifications. Receiving notifications about such a delicate and harmful subject can be shocking for people, especially when involving teenagers or fragile people, and could potentially cause harm and dangerous reactions. The choices regarding how the system has to deliver the notification, who has to receive it (users or their parents?), and the best moment to send/receive it, should be made by an expert (e.g., a psychologist). Indeed, as already mentioned, this is an interdisciplinary topic and, although we here limit our investigation to the technological tools that could be used to limit non-consensual pornography, experts in other fields should be involved as well before deploying such a system.

6.3 Classification of private/sensitive images

The simplest version of a restricted forwarding algorithm is the one that simply applies the forwarding policy to any image generated by someone else, no matter how harmless. However, this would have an impact on effectiveness and scalability. Let us consider the case where user A sends out an image of a public place (or a celebrity, some piece of art, a monument, a flyer of an event, etc.) to a user B. Then, if user C tries to send out the same image to user D, she/he will find out that the system triggers the procedure specified by the forwarding policy, even though the photo is not a private/sensitive one. Therefore, considering all the images as susceptible to forwarding control is clearly excessive.

Triggering the behavior specified by the forwarding policy when not necessary can be a non-trivial problem for users and their experience, which could lead them not to use our application. As a consequence, *SafeSext* would not reach its goal. Moreover, since the performances of our forwarding



Hi 10:31

Hi 👋 10:31

Sexually Explicit Image 10:32

Hi 10:31

Hi 👋 10:31

Sexually Explicit Image 10:32

Hi 11:03

Sexually Explicit Image 11:03

Attention

You are not allowed to send this photo as you are not the owner

(a) M sends a pornographic image to A.

(b) A receives the image from M.

(c) A tries to send the image to someone else, but the system blocks the operation and shows an alert.

Fig. 8. Example of block of the sending.

algorithm depends on the number of saved hash values, the more images the system considers relevant, the bigger this number is, thus jeopardizing scalability. So, we analyzed and tested other possible solutions.

The first solution we considered was to let the users choose which images should be protected by the forwarding control policies. For instance, users could enable the restriction by clicking on a button on the image preview before sending it out. However, such a solution works only in an ideal scenario, where users exclusively select personal photos. Instead, problems arise when users enable the forwarding restrictions on images taken, for instance, during a vacation. Several pictures of the same monument taken from similar locations and angles would result as a violation of the forwarding restriction even if the images have been generated by a different user. Consider, for instance, the millions of pictures taken every year of the Colosseum in Rome or of the Tour Eiffel in Paris: only the first tourist attempting to send such pictures would be enabled to do so, blocking all other tourists from there on. Unfortunately, soon all the images would have an owner, jeopardizing the system and bringing back the aforementioned usability drawbacks. We hence discarded this approach.

Therefore a solution can be to change the approach and ask the system to identify which images can be freely shared, i.e., pictures of public places, and to consider private the others. For this reason, we test the functionalities for landmarks detection of the Google Cloud Vision API against a dataset composed of some images downloaded from the Internet. We hypothesized that this API was able to correctly identify pictures depicting some famous monuments, but not general public places (e.g., roads, mountains, beaches, etc.). Hence, to verify such a hypothesis, our dataset contained images of both kinds and photos of people (i.e., the negative part). Unfortunately, the API was able to recognize only about 1 over 3 pictures containing monuments/public places (the True Positive Rate is 0.31). Even if it can correctly classify the negative portion of our dataset (the True Negative Rate is 0.96), this solution must be discarded because photos which do not contains famous monuments but common public places, e.g., a road, are classified in the wrong way. This is a consequence of the difference between the concepts of *monument* and *public place*. Moreover, such a solution would not permit the detection of images depicting celebrities or some piece of art.

The last considered approach, which is the one actually implemented in *SafeSext*, imposes restrictions only on the necessary images, recognizing the ones that contain sexual contents. In this way, our system can restrict the forwarding only to a limited set of images, preserving the user experience and minimizing the drawbacks described above. To this aim, our system uses the feature for detecting explicit content of the Google Cloud Vision API, which we tested against two different datasets. The first dataset is a subset of the one proposed in [38], which contains pornography images and non-pornography ones (both easy and difficult to detect), while the second one is composed of some softer pornography pictures and some photos depicting people downloaded from some social networks. In the first test, we evaluated the API against 1000 images from each category

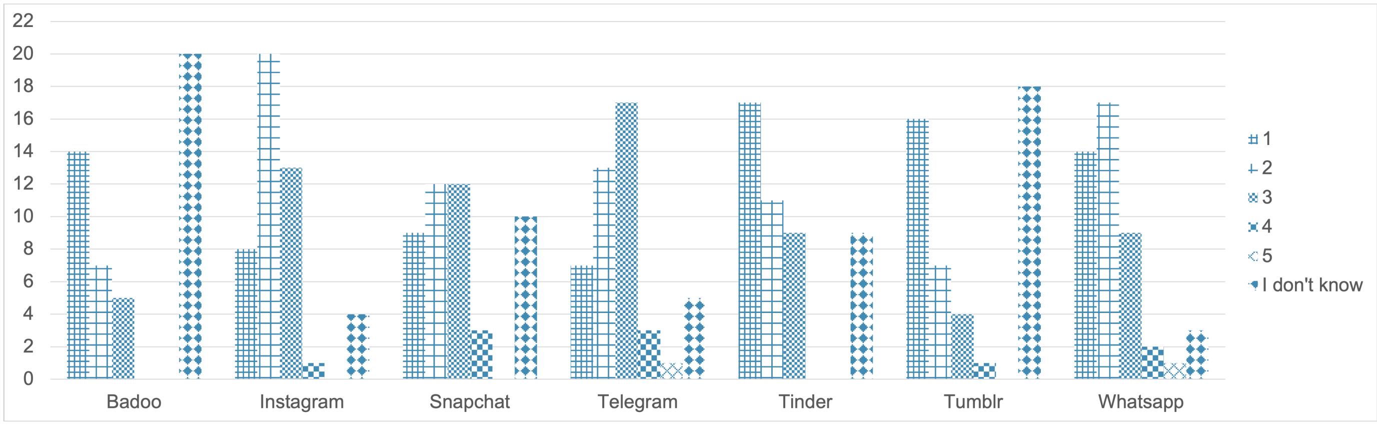


Fig. 9. Users' evaluation of the analyzed applications, from 1 (*very little*) to 5 (*very much*).

of the dataset contents proposed in the NPDI Pornography Database [38]. The result for the pornography category is excellent: the classification is correct for all the examples. The performance gets worse with the easy non-pornography pictures where the True Positive Rate is 0.81, which is still acceptable. Instead, the result reached by the difficult category is a True Positive Rate of 0.36, which is unacceptable. The dataset for the second test was composed of 373 softporn images downloaded from public Tumblr profiles and 227 pictures of people. The API under test works very well against this dataset. Indeed, we obtained a True Positive Rate equal to 0.97 and a False Positive Rate of 0.45. These values highly depend on the definition of what is relevant for the system. In our case, the negative samples contain many images depicting people in beachwear (e.g., bikinis, swimwear, etc.) that are often recognized as adult content by the Google Safe Search API, leading to a high False Positive Rate. However, other definitions of personal content may be considered. For instance, users in Arab regions may have a more strict notion of personal and sensitive content (for instance including naked shoulders) with respect to Western users [39]. Therefore, 0.45 is a worst-case value and can decrease drastically, even close to 0, considering pictures depicting people wearing beachwear as positives and/or employing a more accurate content moderation API (e.g., Amazon Rekognition).

Considering the aforementioned results, our analysis shows that imposing restrictions only on images containing sexual content is the most viable approach amongst the considered ones. In this way, our system stores only the strictly necessary hash values, improving the scalability and the user experience while reducing the chance of error.

6.4 Updating the Fingerprinting Function

With technical and scientific advancements, the performance of current systems can continue to improve and provide the best possible experience to their users. On the other hand, developers have to design systems that are easy to maintain and update, possibly with transparent procedures from the user standpoint. In this way, developers and system administrators can update their systems, minimizing disservices for users, which is particularly important for all the essential platforms or those with a minimum set of services that need to be available in nearly any scenario.

Clearly, a messaging system is not a life-critical service, but when one goes down for some issues, people move *en masse* to other platforms, with an economic loss for the company loosing customers. In our case, it is easily foreseeable that the perceptual hashing function will have to be updated every time a more performing one will be available, especially considering that the problem of image authentication is an adversarial one. However, a messaging platform cannot suspend its service every time its perceptual hashing function needs an update. Thereby, we designed a transparent procedure to reach this goal.

Such functionality justifies the presence of private storage for images on the backend server. Indeed, such a choice comes with some privacy and legal concerns. After investigating how popular messaging platforms store messages and pictures, we can claim that this generally done in the field by main companies. The main issue with this solution is that, even with symmetric encryption, we can see the images stored in our system, which could potentially be a serious concern from the user point of view, especially considering the private content our system handles. Moreover, underage users can send sexy photos of themselves, causing our system to store child pornography content, which is illegal in most countries. For these reasons, another possibility is to explicitly ask users to (temporarily) upload their relevant images every time the fingerprinting function has to be updated. Clearly, if someone does not accept or does not have access to Internet connectivity, their pictures will no more be checked by the (updated version of the) forwarding control algorithm until they are uploaded. In this way, the system does not need to store any image on servers, further preserving users' privacy.

To update the fingerprinting function, developers can access a private website to load the new fingerprinting model into the system, starting the update. Our procedure turns off the forwarding control functionality for the necessary time while the rest of the system continues to work regularly. Then, each hash value is computed again through the new fingerprinting function by using the images stored in the private storage (or temporarily uploaded by users in case of private storage). Finally, the system turns on its forwarding control feature again.

7 Users Assessment of the proposed solution

Besides the performance evaluation of our system discussed in Section 6.1.1 we asked 46 people in the age group from 20 to 55 to evaluate *SafeSext*, through an online questionnaire. In particular, participants were asked about the proposed forwarding policies and the necessity of detecting also edited copies of an image. They were asked to evaluate all answers on a Likert scale from *1 (very little)* to *5 (very much)*. Before starting the questionnaire, participants, who were recruited among the master's students of Computer Science at the University of Padua and their contacts, were informed about the topic of the survey with a brief presentation. Because of the intimate nature of the questions and the sensitive topic, they could decide not to send their answers and terminate the questionnaire at any moment. In addition, to create the most comfortable environment, they could complete the questionnaire in the context they preferred (e.g., home, etc.). No identifiable data were collected, and the answers were completely anonymous.

Participants were approximately equally distributed between females and males, respectively 22 and 24. 23 of them (50%) are aged 20 to 25, 17 (37%) are from 26 to 35, and 6 (13%) participants are aged between 36 and 55. More than 85% of them had some knowledge about sexting, about 90% of participants confirmed that they did sexting at least once in their life, and 35% of them have done sexting with 4 or more people during their life.

We started by asking to express an evaluation of safeness of the most popular messaging applications. In Fig. 9 we report the answers to the question "*How confident would you feel in terms of feeling protected from the abuse of self-generated sexual content?*" using Badoo, Instagram, Snapchat, Telegram, Tinder, Tumblr or Whatsapp. As it is clear from the outcome, participants do not trust these applications, even though there is some heterogeneity in the evaluation. Furthermore, sometimes the perceived safeness of the system is not coherent with the actual features provided by the considered platform. For instance, Tinder has a very low rating despite it does not permit sending media at all; so it could be considered, by design, the safest one.

In Fig. 10, we reported the answers to the question "*How much do you agree with the following statement: Any messaging application should implement the following forwarding control policy*". We did not ask to participants to compare the two policies but only to express their agreement on the usefulness of each single policy in comparison with nothing at all. The responses show that both the policy which blocks the forwarding and the forward with notification are claimed to be useful features, even though the former received 33% more selections of the option *5 (very much)* and therefore is considered more appropriate. This is expected with users exploiting the application for safe sexting as a well designed and performing blocking system (even in combination with a notification) is certainly safer than one which permits to forward images in any case.

Furthermore, we asked to the participants to assess the importance of detecting edited copies of a private image ("*How important is to detect not only the original media but also copies edited to modify them?*"). Fig. 11 shows the answer to this question: about 60% of participants have selected option *5 (very much)* and no one selected options *1* and *2*.

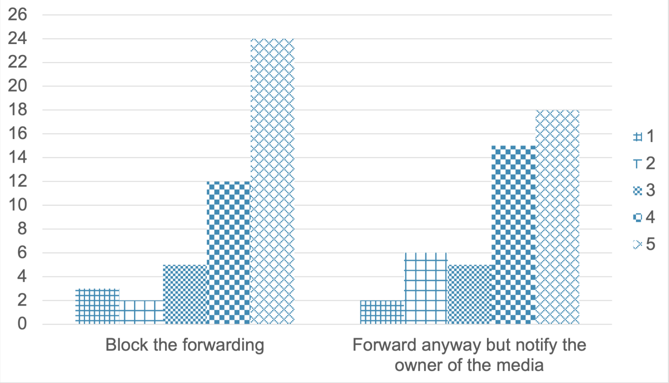


Fig. 10. Assessment of the proposed forwarding policies, from 1 (*very little*) to 5 (*very much*).

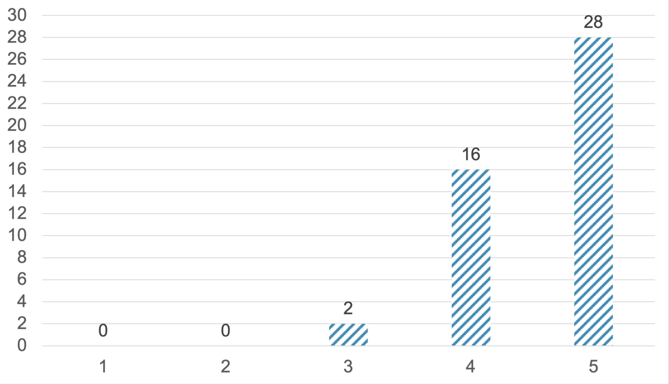


Fig. 11. Assessment of the importance of detecting even edited images, from 1 (*very little*) to 5 (*very much*).

8 Conclusion

Sexting has gained popularity amongst teenagers and young adults; yet, it can have serious consequences, such as non-consensual pornography through the leak of private and sensitive media without the owner's consent [13], [14]. Furthermore, commonly used messaging applications do not limit such a drawback since they permit sending any content to anyone. In essence, they are not safe enough for sexting [22].

In this paper, we have proposed a proof of concept of a messaging system that aims at increasing the safety of users engaging in sexting thanks to an image forwarding control feature. We discussed challenges, possible solutions and limits, thus defining a research agenda in this field. We have also demonstrated that social media platforms may be considered co-responsible for issues related to sexting abuse since they could actually adopt possible technological countermeasures to avoid them.

The purpose of our system is indeed to prevent the uncontrolled spreading of personal self-generated nude or semi-nude pictures without the owner's consent, reducing well-known problems to which their dissemination can lead. In this context, we have designed a forwarding control algorithm and tested each of its parts, including its time complexity. We have also trained a neural network to compute a perceptual hashing function for images, evaluating how its performances influence effectiveness and user experience. Furthermore, to improve performance and interaction with our platform, we have discussed which images our algorithm has to consider relevant and how to automatically recognize them. We have then presented some possible policies (i.e. behaviors) to use when the algorithm detects suspicious forwardings and asked the users their opinion on them.

Future Research Directions. This work also aims at opening new research directions, showing that further work is needed in this context on different problems, requiring heterogeneous expertise. For instance, implications on the security and privacy of our system need further investigation, even considering the compliance to European regulations and United States law (e.g., GDPR, DMCA).

The fingerprinting function deserves a more extensive analysis, ranging from the image manipulations and their impact in a real scenario to the improvement of its performance (e.g., reduction of false positives and false negatives). We must note here that we selected on purpose image transformations widely available on image editing applications and easy to be used. Yet, algorithms for a broad part of the possible manipulations are proprietary, or a commonly accepted definition could not exist. For instance, this is the case of brightness, which could have a relevant impact on the performance of the hashing function in a real scenario and deserves further investigation in the future. In addition, an exploration of perceptual hashing functions for other kinds of media would make *SafeSext* more comprehensive.

To detect whether an image is relevant for the system or not, Google Cloud Vision API is used. Other APIs could be considered, even solutions that work locally or without employing external services, thus improving the privacy level of the system. Moreover, the management of screenshot attempts and screen recordings needs further research, even considering the differences between devices and operating systems. Indeed, they have a disruptive effect on our system. Yet, only the Android OS allows developers to disable the possibility of taking screenshots. On the other hand, iOS notifies the applications when the user takes one (whereas, in Android, there are only some not-official solutions).

The effectiveness of our system needs also to be further investigated from a human-centred perspective. For instance, understanding the motivation behind the numeric evaluations of the forwarding policies provided by participants to our survey would be interesting.

Acknowledging the importance of the topic, we would also like to integrate our platform with the National Center for Missing & Exploited Children's services (NCMEC) to detect the content of child exploitation and abuse, also providing users with features to report such material. Furthermore, the minimum age above which people can use our application needs to be discussed, especially if we consider that sexting-related images depicting underage people are illegal in many countries.

Finally, involving teenagers and experts in other fields (e.g., sexual health experts, psychologists, etc.) during the design of a messaging system would be particularly useful in creating an application ready to be actually deployed.

Acknowledgments

This work is partially funded by the Department of Mathematics of the University of Padua through the BIRD191227 project.

References

[1] S. Counts and K. E. Fisher, "Mobile social networking as information ground: A case study," *Library & Information Science Research*, vol. 32, no. 2, pp. 98–115, 2010.

[2] Y. Barrense-Dias, A. Berchtold, J.-C. Surís, and C. Akre, "Sexting and the definition issue," *J. of Adolescent Health*, vol. 61, 07 2017.

[3] E. C. Stasko and P. A. Geller, "Reframing Sexting as a Positive Relationship Behavior," https://www.apa.org/news/press/releases/2015/08/reframing-sexting.pdf, 2015, online; accessed 05 April 2022.

[4] N. Henry, A. Powell, and A. Flynn, "Not just 'revenge pornography': Australians' experiences of image-based abuse. a summary report," *RMIT University*, 2017.

[5] Y. Ruvalcaba and A. A. Eaton, "Nonconsensual pornography among u.s. adults: A sexual scripts framework on victimization, perpetration, and health correlates for women and men," *Psychology of Violence*, vol. 10, no. 1, pp. 68–78, 2019.

[6] S. Bates, "Revenge porn and mental health: Qualitative analysis of the mental health effects of revenge porn on female survivors," *Feminist Criminology*, pp. 1–21, 2016.

[7] P. Wisniewski, A. K. Ghosh, H. Xu, M. B. Rosson, and J. M. Carroll, "Parental control vs. teen self-regulation: Is there a middle ground for mobile online safety?" in *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, ser. CSCW '17. Association for Computing Machinery, 2017, p. 51–69.

[8] A. K. Ghosh, K. Badillo-Urquiola, S. Guha, J. J. LaViola Jr, and P. J. Wisniewski, "Safety vs. surveillance: What children have to say about mobile apps for parental control," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, ser. CHI '18. Association for Computing Machinery, 2018, p. 1–14.

[9] B. McNally, P. Kumar, C. Hordatt, M. L. Mauriello, S. Naik, L. Norooz, A. Shorter, E. Golub, and A. Druin, "Co-designing mobile online safety applications with children," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, ser. CHI '18. Association for Computing Machinery, 2018, p. 1–9.

[10] A. K. Ghosh, C. E. Hughes, and P. J. Wisniewski, "Circle of trust: A new approach to mobile online safety for families," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, ser. CHI '20. Association for Computing Machinery, 2020, p. 1–14.

[11] N. Döring, "Consensual sexting among adolescents: Risk prevention through abstinence education or safer sexting?" *Cyberpsychology: Journal of Psychosocial Research on Cyberspace*, vol. 8, no. 1, p. Article 9, Mar. 2014. [Online]. Available: https://cyberpsychology.eu/article/view/4303

[12] P. Korenis and S. Billick, "Forensic implications: Adolescent sexting and cyberbullying," *The Psychiatric quarterly*, vol. 85, 10 2013.

[13] A. Razi, K. Badillo-Urquiola, and P. J. Wisniewski, "Let's talk about sext: How adolescents seek support and advice about their online sexual experiences," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, ser. CHI '20. Association for Computing Machinery, 2020, p. 1–13.

[14] H. Hartikainen, A. Razi, and P. Wisniewski, "Safe sexting: The advice and support adolescents receive from peers regarding online sexual risks," *Proc. ACM Hum.-Comput. Interact.*, vol. 5, no. CSCW1, apr 2021.

[15] X. Wang, C. Hu, and S. Yao, "An adult image recognizing algorithm based on naked body detection," in *2009 ISECS International Colloquium on Computing, Communication, Control, and Management*, vol. 4, 2009, pp. 197–200.

[16] C. Santos, E. M. dos Santos, and E. Souto, "Nudity detection based on image zoning," in *2012 Conference on Information Science, Signal Processing and their Applications (ISSPA)*, 2012, pp. 1098–1103.

[17] H. Sevimli, E. Esen, T. K. Ateş, E. C. Ozan, M. Tekin, K. B. Loğoğlu, A. M. Sevinç, A. Saracoğlu, A. Yazici, and A. A. Alatan, "Adult image content classification using global features and skin region detection," in *Computer and Information Sciences*, E. Gelenbe, R. Lent, G. Sakellari, A. Sacan, H. Toroslu, and A. Yazici, Eds. Dordrecht: Springer Netherlands, 2010, pp. 253–258.

[18] M. U. Tariq, A. Razi, K. Badillo-Urquiola, and P. Wisniewski, "A review of the gaps and opportunities of nudity and skin detection algorithmic research for the purpose of combating adolescent sexting behaviors," in *Human-Computer Interaction. Design Practice in Contemporary Societies*, M. Kurosu, Ed. Cham: Springer International Publishing, 2019, pp. 90–108.

[19] A. Razi, S. Kim, A. Alsoubai, G. Stringhini, T. Solorio, M. De Choudhury, and P. J. Wisniewski, "A human-centered systematic literature review of the computational approaches for online sexual risk detection," *Proc. ACM Hum.-Comput. Interact.*, vol. 5, no. CSCW2, oct 2021. [Online]. Available: https://doi.org/10.1145/3479609

[20] M. Wood, G. Wood, and M. Balaam, "Sex talk: Designing for sexual health with adolescents," in *Proceedings of the 2017 Conference on Interaction Design and Children*, ser. IDC '17. Association for Computing Machinery, 2017, p. 137–147.

[21] V. Guana, T. Xiang, H. Zhang, E. Schepens, and E. Stroulia, "Undercontrol an educational serious-game for reproductive health," in *Proceedings of the First ACM SIGCHI Annual Symposium on Computer-Human Interaction in Play*, ser. CHI PLAY '14. Association for Computing Machinery, 2014, p. 339–342.

[22] M. Franco, O. Gaggi, and C. E. Palazzi, "Improving sexting safety through media forwarding control," in *2022 IEEE Consumer Communications & Networking Conference (CCNC)*, 2022, pp. 1–6.

[23] S. Wachs, M. F. Wright, M. Gámez-Guadix, and N. Döring, "How are consensual, non-consensual, and pressured sexting linked to depression and self-harm? the moderating effects of demographic variables," *International Journal of Environmental Research and Public Health*, vol. 18, no. 5, 2021.

[24] "How Sexting Helped Me Embrace My Disabled Body," https://femsplain.com/how-sexting-helped-me-embrace-my-disabled-body-bc33833f7f88, 2015, online; accessed 04-11-2021.

[25] M. Furini, S. Mirri, M. Montangero, and C. Prandi, "Privacy perception and user behavior in the mobile ecosystem," in *Proceedings of the 5th EAI International Conference on Smart Objects and Technologies for Social Good*, ser. GoodTechs '19, 2019, p. 177–182.

[26] M. Furini, S. Mirri, M. Montangero, and C. Prandi, "Privacy perception when using smartphone applications," *Mobile Networks and Applications*, vol. 25, no. 3, 2020.

[27] M. K. Mıhçak and R. Venkatesan, "New iterative geometric methods for robust perceptual image hashing," in *Security and Privacy in Digital Rights Management*, T. Sander, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 13–21.

[28] B. Yang, F. Gu, and X. Niu, "Block mean value based image perceptual hashing," in *2006 International Conference on Intelligent Information Hiding and Multimedia*, 2006, pp. 167–172.

[29] Z. Tang, Y. Dai, and X. Zhang, "Perceptual hashing for color images using invariant moments," *Applied Mathematics and Information Sciences*, vol. 6, pp. 643S–650S, 04 2012.

[30] V. Monga, A. Banerjee, and B. Evans, "A clustering based approach to perceptual image hashing," *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 1, pp. 68–79, 2006.

[31] L. Du, A. T. Ho, and R. Cong, "Perceptual hashing for image authentication: A survey," *Signal Processing: Image Communication*, vol. 81, p. 115713, 2020.

[32] C. Qin, E. Liu, G. Feng, and X. Zhang, "Perceptual image hashing for content authentication based on convolutional neural network with multiple constraints," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 11, pp. 4523–4537, 2021.

[33] Y. Li, D. Wang, and L. Tang, "Robust and secure image fingerprinting learned by neural network," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 2, pp. 362–375, 2020.

[34] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 740–755.

[35] P. Nyoni and M. Velempini, "Privacy and user awareness on facebook," *South African Journal of Science*, vol. 114, no. 5/6, p. 5, May 2018. [Online]. Available: https://sajs.co.za/article/view/5165

[36] B. Henne and M. Smith, "Awareness about photos on the web and how privacy-privacy-tradeoffs could help," in *Financial Cryptography and Data Security*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 131–148.

[37] C. Liu, T. Zhu, J. Zhang, and W. Zhou, "Privacy intelligence: A survey on image privacy in online social networks," *ACM Comput. Surv.*, jul 2022, just Accepted. [Online]. Available: https://doi.org/10.1145/3547299

[38] S. Avila, N. Thome, M. Cord, E. Valle, and A. de A. Araújo, "Pooling in image representation: The visual codeword point of view," *Computer Vision and Image Understanding*, vol. 117, no. 5, pp. 453–465, 2013.

[39] N. Abokhodair and S. Vieweg, "Privacy & social media in the context of the arab gulf," in *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*, ser. DIS '16. Association for Computing Machinery, 2016, p. 672–683.

Mirko Franco is a PhD Student in Brain, Mind and Computer Science at the Department of Mathematics of the University of Padua, under the supervision of Prof. Claudio E. Palazzi. He previously completed his B.Sc. degree and M.Sc. degree in Computer Science at the same university, respectively, in 2019 and 2021. His current research activity mainly focuses on mobile systems and social platforms and is involved in the organization of conferences such as IEEE NES and ACM GoodIT.

Ombretta Gaggi is an Associate Professor in Computer Science at the Department of Mathematics of the University of Padua. She received her M.S. degree in Computer Science from University of Venice in 1998 and her Ph.D. degree in Computer Science from University of Bologna in 2003. Her research interests include accessibility and web technologies, cross-platform mobile development and user interface design. She is member of several technical program committees of international conferences and of the steering committee of ACM GoodIT and associate editor of Multimedia Tools and Applications Journal.

Claudio E. Palazzi is an Associate Professor in Computer Science at the Department of Mathematics of the University of Padua. He received his M.S. degree in Computer Science from UCLA in 2005, his Ph.D. degree in Computer Science from UniBO in 2006, and his Ph.D. degree in Computer Science from UCLA in 2007. His research interests are primarily focused on the design and analysis of Internet architectures and mobile systems. He is member of the steering committee of conferences such as IEEE CCNC, ACM GoodIT and IFIP/IEEE Wireless Days and associate editor of IEEE Transactions on Multimedia and Elsevier Computer Networks.