A Privacy-Preserving Method for Photo Sharing in Instant Message Systems

Fenghua Li Jingyang Yu Lingcui Zhang* Zhe Sun Mengfan Lv
State Key Laboratory of Information Security
Institute of Information Engineering, Chinese Academy of Sciences
Beijing, China
{lifenghua, yujingyang, zhanglingcui, sunzhe, lvmengfan}@iie.ac.cn

ABSTRACT
In instant messaging systems, photos have become an expressive form of real-time content sharing among users. Unfortunately, faces of stakeholders, who are depicted in the shared photos, may be exposed to unexpected viewers. This may lead to privacy leakages. In this paper, an approach is proposed to prevent such leakages, which is based on access control and face recognition. Each time a photo is sent, all stakeholders are recognized, and their faces are hidden from viewers. At the same time, stakeholders will be notified about the photo, and can decide to reveal their own faces to some proper viewers later. On the other hand, similar photos are exploited to automate this revealing process when possible, so as to reduce the impact of access control on user experiences. Besides, a proof-of-concept system is constructed and its performance is evaluated. The results indicate that this approach can be applied in instant messaging systems without too much overhead.

CCS Concepts
• Security and privacy~Social network security and privacy

Keywords
Instant messaging; Photo sharing; Privacy; Face recognition.

1. INTRODUCTION
Instant messaging (IM) systems allow efficient communication among their users. With the rapid growth of Mobile Internet, users can exchange messages almost at any time or place. Therefore, IM applications such as WeChat and WhatsApp have become popular nowadays. Furthermore, the supports for multimedia messages, especially photos, can greatly enrich the shared contents. WeChat had about 600 million users by 2014, 70.8% of whom send photo messages to their friends [1]. Moreover, the number of users has reached about 800 million in 2016, as is shown in [2].

Unfortunately, there exist some privacy issues during photo sharing [3]. One is that faces of stakeholders (depicted users) may be exposed to unexpected viewers, which has been alarming due to the popularity of IM applications. There have already been some mechanisms in popular IM systems that allow users to hide faces. These mechanisms, however, require that photo uploaders process the photos before sending them. As a result, if multiple stakeholders are depicted in a photo, the exposure of faces can easily get out of control. In such situations, however carefully stakeholders prevent their faces from being exposed, other users can nonetheless send the unprocessed photo, either accidently or intentionally.

ICCSP '17, March 17-19, 2017, Wuhan, China

DOI: http://dx.doi.org/10.1145/3058060.3058081

Some works have already tried to address some privacy issues in IM systems, but mostly focus on the presence mechanisms instead of message contents [4, 5]. On the other hand, a large number of approaches have been proposed to prevent privacy leakages from photos in web-based OSNs (online social networks). These approaches concentrate on different aspects of access control, such as policy prediction, conflict resolution, and subject identification. Although they are enlightening for addressing similar problems in IM systems, some differences between the two kinds of systems have rendered them difficult to apply. Therefore, an approach is needed to deal with problems that are specific to IM systems.

In this paper, an approach that is suitable for IM systems is proposed. Stakeholders can decide the visibilities of their own faces regardless of whether they are the photo senders. Besides, similar photos are exploited to automate this decision making process, since user experiences will be influenced by late decisions.

The primary contributions of this work are:

- A privacy-preserving approach for photo sharing in IM systems, which allows stakeholders to decide whether their faces should be presented to some viewers.
- Automated access control by exploiting similar photos, which makes decisions for users whenever possible, thus helping preserve user experiences.
- A proof-of-concept implementation, which shows that this approach integrates well with IM systems.

The remainder of this paper is organized as follows. Section 2 describes the related work. Section 3 provides a detailed explanation of the proposed system. Some implementation details and the results of some performance tests are given in section 4. Section 5 has a discussion of the limitations and possible solutions. In section 6, the main conclusions are established.

2. RELATED WORK
Privacy-preserving photo sharing has been an active research field, and the rise of web-based OSNs such as Facebook has been one of the main driving forces. Since photos are an important means of

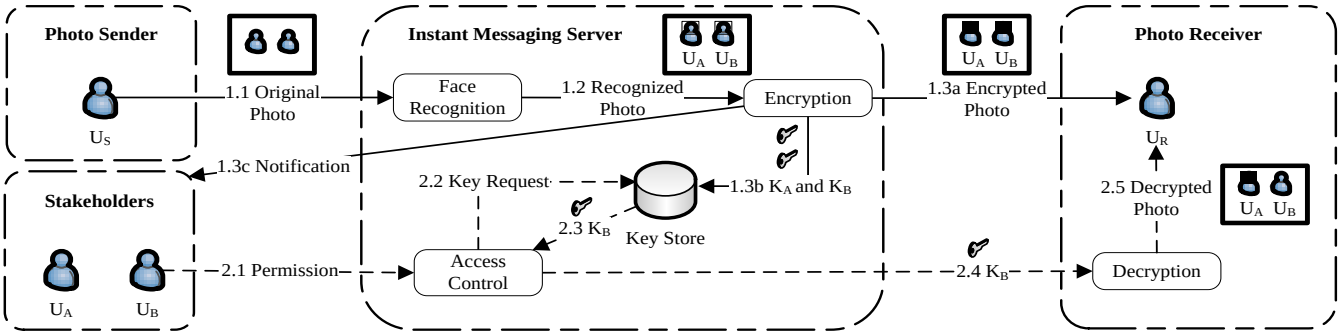


Figure 1. The system overview.

content sharing in web-based OSNs, it is inevitable that some privacies will be leaked from the uploaded photos. Therefore, many works have been concentrating on web-based OSNs.

In web-based OSNs, when users upload photos, they can set the visibilities of the photos for different contacts. Some works focus on the prediction of such policies. According to the work of Squicciarini et al., policies can be predicted basing on photo classification and historical policies [6]. Kairam et al. have shown that aesthetic properties of photos can also play an important role [7]. While in [8], Ni et al. discussed the utilization of user profiles as well as some temporal and spatial information for prediction. However, in these approaches, only the photo uploader controls the policies. At the same time, the visibilities are controlled at the photo level, so it is not possible to hide only part of a photo.

In [9, 10], Hu et al. has proposed approaches for stakeholders to participate in the access control. Stakeholders can be identified by tags on the photos. As a result, they can get notified when photos of them are uploaded, and can subsequently decide proper viewers of the photos. The final access control policy is a mix of all stakeholders' policies. Palomar et al. presented a system where users have attributes and the visibilities of photos are controlled by setting attribute restrictions [11]. These approaches also control visibilities at the photo level, so policy conflicts may arise. In the meantime, an uploader needs to tag all the users in photos correctly, while this mechanism is not available in an IM system.

Xu et al. introduced face recognition into web-based OSNs, so as to identify the stakeholders and let them decide the access control policies collaboratively [12]. Still, visibilities are controlled at the photo level, so policy conflicts remain. Ilia et al. changed the granularity of visibilities control to the face level, and stakeholders can only control their own faces [13]. As a result, the conflicts are resolved naturally. However, this approach does not address some challenges faced by IM systems. One is that a viewer will not fetch the same photo repeatedly, so the visibility changes of the photo cannot be easily perceived. Another is that photos in IM systems are much more time-sensitive, so access control will have an impact on user experiences.

3. SYSTEM DESIGN

In popular IM systems, the sender is usually the only one who can decide whose faces in a photo should be hidden. However, senders will not always consider stakeholders' privacy and may expose their faces to others. While in this proposed system, stakeholders are able to control the visibilities of their own faces.

An overview of the proposed system is depicted as Figure 1. As is shown, this approach consists of two workflows. When sending a photo, the stakeholders' faces will be encrypted before the photo is actually sent to the receivers. This workflow is described in section 3.1. When a stakeholder decides the proper viewers of the photo afterward, the keys for decryption will be sent to these viewers. This workflow is described in section 3.2.

3.1 Photo Sending

Algorithm 1 presents the process of sending a photo from the sender to the receivers. The whole process is divided into three steps as is commented. Each of the three following paragraphs will have a detailed description of one of the steps.

The first step is face recognition. When a photo arrives at the server, face detection is performed to find all faces depicted in it. Then user identification is performed on the detected faces to find the stakeholders. Users need to upload their own face images in advance if they want themselves to be identified, while the server extracts face data from these images and saves it to a database. During user identification, this database is queried to find the most likely owner of a detected face, as well as the corresponding confidence. Only if the confidence exceeds a predefined threshold, will the found user be accepted as a stakeholder.

The second step is face hiding. For each recognized stakeholder, the corresponding face region is encrypted with a randomly generated key. Then an allow list including all proper viewers of the face is generated. This list will be the same as the stakeholder's default allow list, which needs to be specified by the stakeholder beforehand. On the other hand, if the photo sender is also a stakeholder, the corresponding allow list will contain the photo receivers as well, since it can be considered that the sender has already qualified the receivers when sending the photo.

The final step is message dispatching. Since photos can be sent to chat rooms, there can be more than one photo receivers. For each receiver that is also a stakeholder, the unprocessed photo can be sent directly. Otherwise, the encrypted photo should be sent. Besides, if the receiver is included in some stakeholders' allow lists, all these stakeholders' keys should be sent together with the photo for decryption. In the meantime, all stakeholders will be notified that a photo of them is sent.

During this workflow, some information needs to be persisted on the server for use in the other workflow. This information includes the stakeholders, their keys and allow lists, the receivers, and their devices used while receiving the encrypted photo.

3.2 Access Control

Default allow lists are used to decide proper viewers of stakeholders' faces. In general, these lists should be restrictive to

Algorithm 1. SendPhoto(*sender*, *target*, *photo*)

```
 1: {Comment: the face recognition step}
 2: S ← recognized stakeholders in photo
 3: {Comment: the face hiding step}
 4: photo_enc ← copy of photo
 5: R ← receivers of photo indicated by target
 6: for all s in S do
 7:   keys[s] ← random generated key
 8:   encrypt s's face in photo_enc with key[s]
 9:   allow_lists[s] ← s's default allow list
10:   if sender = s then
11:     allow_lists[s] ← allow_lists[s] ∪ R
12:   end if
13: end for
14: {Comment: the message dispatching step}
15: for all r ∈ R do
16:   devices[r] ← r's current device
17:   if r ∈ S then
18:     send photo to devices[r]
19:   else
20:     send photo_enc to devices[r]
21:     S' ← {s ∈ S | r ∈ allow_list[s]}
22:     for all s ∈ S' do
23:       send keys[s] to devices[r]
24:     end for
25:   end if
26: end for
27: for all s ∈ S do
28:   notify s
29: end for
```

prevent faces from being exposed to too many users. However, since each photo has its own set of proper viewers, stakeholders should have chances to change the allow lists for a photo. In the proposed system, as long as stakeholders have already received the notifications, they can change the lists anytime they want. Then faces in the photos that are already sent to receivers will be patched.

Algorithm 2 presents the server side work for changing an allow list. The new list is compared with the old list to find the differences. On the one hand, for each added viewer who has received the encrypted photo, the stakeholder's key for the photo should be sent. On the other hand, for each removed viewer who has received the photo, a request should be sent to delete the stakeholder's key. When sending keys or delete requests, the server needs to ensure that the viewer is using the same device as the one used to receive the encrypted photo. Device IDs could be used to accomplish this.

Upon receiving a key, an IM client can decrypt the corresponding face region for the viewer. However, the client should not persist the decrypted photo on the local storage, so that when a delete request is received, it can simply delete the key to prevent the face from being viewed.

This workflow requires that the system support offline messages and chat histories. In this manner, even if the viewer is offline, or was once offline, a photo that is already sent to the viewer can be properly patched. Besides, with support for chat histories, device IDs can be generated by the server and written into chat histories by the clients. Then when logging in, a client can tell the server the saved device ID as soon as it has loaded the chat history. This can greatly simplify chat history operations.

Algorithm 2. ChangeAllowList(*stakeholder*, *new_list*)

```
1: old_list ← allow_lists[stakeholder]
2: allow_lists[stakeholder] ← new_list
3: R ← memorized receivers of photo
4: for all r ∈ (new_list – old_list) ∩ R do
5:   send keys[stakeholder] to devices[r]
6: end for
7: for all r ∈ (old_list – new_list) ∩ R do
8:   request devices[r] to delete keys[stakeholder]
9: end for
```

3.3 Policy Reuse

Face hiding can have a great impact on the user experience. For any photo exchanged in the system, only after the stakeholders grant permissions, faces can be shown. However, it can take quite a while for the stakeholders to take actions. For example, a photo can be sent while a stakeholder is offline. When the stakeholder logs in and grants the permission later, the viewer may have lost interest in the original photo.

Therefore, the access control process needs to be automated. One simple method is to use default allow lists predefined by users, which has already been covered in Figure 2. A more complicated method is to exploit similar photos for policy reuse. There exist many similar photos in the system. For example, some photos contain almost the same content, except that they differ slightly in camera settings, camera positioning, or content arrangement. These photos can share the same allow list. Even if the differences are not slight enough to share the same list, a recommendation can be made basing on a previously used list.

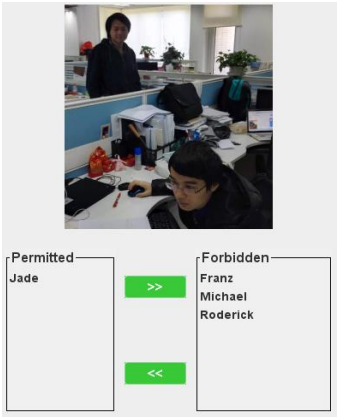
A measurement of similarity is needed for such a method. In the proposed system, similarity measuring is based on local features in photos, so that slight changes of photo properties will not have great influence on the result. A dictionary (BOW, bag of words) consisting of some representative features need to be built beforehand. Then for each extracted feature, the nearest feature in the dictionary is found. Next, a histogram for the frequencies of the representative features is constructed. Finally, the similarity of two photos can be measured in terms of a variety of histogram similarity measurements such as included angle cosine.

Basing on this measurement, allow lists can be reused. When a photo is relayed by the server, other photos with at least one of the same users depicted are found first, which helps reduce the number of photos to be compared. Then the photo with the maximum similarity is found. If this similarity exceeds a predefined threshold, the two photos are considered effectively the same, and can use the same allow list. Otherwise, the allow list for the found photo is recommended to the stakeholder to help make decisions.

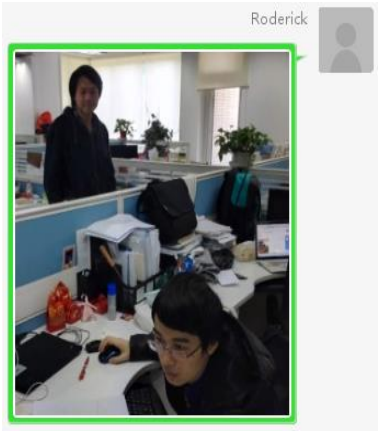
4. EVALUATION

4.1 A Proof-of-concept Implementation

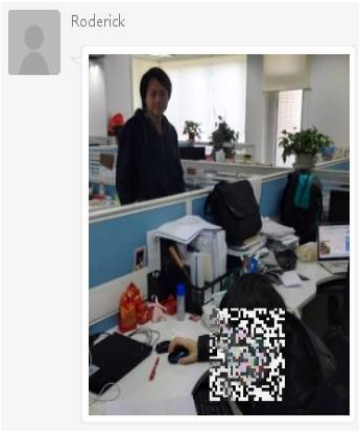
A working IM system is built as a proof of concept. Three screenshots are taken on the client sides when a photo is sent, which are shown in Figure 2. Figure 2(a) shows how the stakeholder grants permissions. The sent photo is placed at the top, while the current allow list is placed at the bottom-left, and candidates for the allow list are listed at the bottom-right. The stakeholder can transfer entries between the two lists with the two buttons indicating directions. The other figures show the effect of face hiding. In Figure 2(b), the sender has sent a photo containing



(a) The stakeholder's view



(b) The sender's view



(c) The receiver's view

Figure 2. Screenshots taken when a photo is sent.

faces of himself and the stakeholder. In Figure 2(c), the receiver has received a photo with the stakeholder's face encrypted.

OpenCV is utilized for face recognition and feature extraction. A pre-trained Haar cascade classifier is used for face detection. LBPH (local binary pattern histogram) algorithm is used for user identification. SIFT (scale-invariant feature transform) algorithm is used to extract the features from photos. First, SIFT is run on 1000 photos to extract features. Next, the k-means algorithm is iterated for 100 times to cluster all extracted features into a dictionary of 1000 features. When a photo needs to be turned into a histogram, the FLANN library is used to find the nearest features in the dictionary. The encryption/decryption algorithms come from the work in [14], but without integrating with the JPEG compression/decompression processes, since it involves some tricky work, which is not necessary for a proof of concept, especially on the client side.

4.2 Performance

For face hiding in real-world IM systems, the performance overhead needs to be taken into account. When a photo is sent from a sender to receivers, several additional steps are involved. Some of these steps are simple database queries. They are not evaluated here since there are other techniques, such as distributed databases, dealing with possible performance issues. Other steps involve image processing, which can be time-consuming. These include face detection, user identification, encryption/decryption, and histogram extraction. Tests are made to evaluate their time consuming. The photos used in the tests come from the MIRFLICKR dataset, except that the ones used for the user identification test come from the LWF (Labeled Faces in the Wild) dataset. For each measured time, 100 randomly selected photos from the dataset are used, and the average time is taken as the final result. A laptop with Intel Core i7-4710MQ and 16GB RAM is used to run the tests.

The average consumed time of these steps is listed in the second column of Table 1, with about 1000 faces in the database. In this implementation, encryption and decryption steps share the same process, so they are tested together. Although the results seem acceptable for IM systems, some parameters are not taken into account. They are listed in the third column of Table 1, and more experiments are made to test their impacts.

The performance overhead of histogram extraction depends on the size of the dictionary. Therefore, a test is run on dictionaries of sizes ranging from 100K to 1000K features, and Figure 3. plots

Table 1. Performance of four steps

Step	Consumed time (ms)	Influential parameters
Face detection	65	-
User identification	374	face database size
Encryption/decryption	37	-
Histogram extraction	155	dictionary size

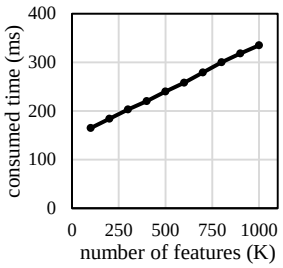


Figure 3. Performance of histogram extraction.

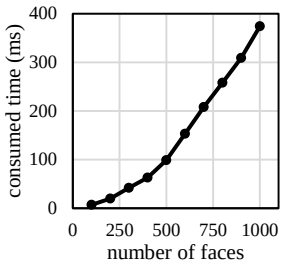


Figure 4. Performance of user identification.

the results. As the results indicate, the average time it takes to extract the histogram increases linearly with the size of the dictionary. However, it is considered affordable, since a dictionary of 1000K features is sufficient in real-world systems, as is indicated in [15].

A simple method for user identification is to compare a given face with all users' face templates to find the best match, which is also how OpenCV implements face recognition. As a result, the performance overhead depends on the number of faces in the face database. Therefore, the test is run on databases of which the number of faces ranges from 100 to 1000, and Figure 4 plots the results. As the results indicate, the time it takes increases significantly as the database size grows. As a consequence, the performance overhead will become unaffordable in a system with a large user base. However, there are some methods that can handle this situation. For example, faces can be classified first, so that when a face needs to be recognized, it needs to be compared with only faces in one class. Besides, according to [16], user identification can be finished in an affordable time.

5. DISCUSSION

This proposed system relies heavily on face recognition. Because the tagging mechanism is absent in IM systems, the accuracy plays a much more important role than in web-based OSNs. So

algorithms like the one in [16] are good choices. However, better algorithms are expected since this research field is still active.

One problem encountered by the system is that even if a face is detected, the corresponding stakeholder may not have an account in the system. Therefore, the larger the user base is, the better users can be protected, which means that this approach can work well in a system like WeChat. For those unidentifiable faces, the server can simply ignore them, or hide them selectively according to other information such as whether the photo seems embarrassing.

6. CONCLUSION AND FUTURE WORK

An approach has been proposed to help protect stakeholders' privacy when photos are exchanged in IM systems. In this system, even if stakeholders cannot prevent senders from sending photos of them, they are given a chance to decide whether to show their faces. An evaluation of performance has shown that this approach can be applied to IM systems without too much overhead. For the future work, there exist several research directions to follow. Firstly, user relationships and chat histories can be utilized to improve the accuracy of face recognition, the importance of which has been covered in section 5. Secondly, more automation on the access control is needed to preserve user experiences. A precise policy prediction method which makes use of photo contents and other information can help with such automation.

7. ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (U1401251), the National High Technology Research and Development Program of China (2015AA016007), the National Natural Science Foundation of China (61672515).

8. REFERENCES

[1] Kuang, W. 2014. Research on the Development of WeChat. *Chinese Journal of Journalism & Communication*, 5 (May 2014), 147-156. DOI= http://dx.chinadoi.cn/10.13495/j.cnki.cjjc.2014.05.011.

[2] Tencent. 2016. Interim Report of Year 2016. http://www.tencent.com/zh-cn/content/ir/rp/2016/attachments/201601.pdf.

[3] Ahern, S., Eckles, D., Good, N., King, S., Naaman, M. and Nair, R. 2007. Over-Exposed? Privacy Patterns and Considerations in Online and Mobile Photo Sharing. In *Proceedings of the ACM Conference on Human Factors in Computing Systems* (San Jose, CA, USA, April 27 - May 3, 2007). CHI '07. ACM, New York, NY, 357-366. DOI= https://doi.org/10.1145/1240624.1240683.

[4] Dumitrache, A., Mileo, A., Zimmermann, A., Polleres, A., Obermeier, P. and Friel, O. 2011. Enabling Privacy-Preserving Semantic Presence in Instant Messaging Systems. In *Proceedings of the International and Interdisciplinary Conference on Modeling and Using Context* (Karlsruhe, Germany, September 26 - 30, 2011). CONTEXT '11. Springer-Verlag, Berlin Heidelberg, 82-96. DOI= https://doi.org/10.1007/978-3-642-24279-3_9.

[5] Kobsa, A., Patil, S. and Meyer, B. 2012. Privacy in Instant Messaging: An Impression Management Model. *Behav. Inform. Technol.*, 31, 4 (Apr 2012), 355-370. DOI= https://doi.org/10.1080/01449291003611326.

[6] Squicciarini, A. C., Lin, D., Sundareswaran, S. and Wede, J. 2015. Privacy Policy Inference of User-Uploaded Images on Content Sharing Sites. *IEEE Trans. Knowl. Data En.*, 27, 1 (Jan. 2015), 193-206. DOI= https://doi.org/10.1109/tkde.2014.2320729.

[7] Kairam, S., Kaye, J. J., Guerra-Gomez, J. A. and Shamma, D. A. 2016. Snap Decisions? How Users, Content, and Aesthetics Interact to Shape Photo Sharing Behaviors. In *Proceedings of the ACM Conference on Human Factors in Computing Systems* (San Jose, CA, USA, May 7 - 12, 2016). CHI '16. ACM, New York, NY, 113-124. DOI= https://doi.org/10.1145/2858036.2858451.

[8] Ni, M., Zhang, Y., Han, W. and Pang, J. 2016. An Empirical Study on User Access Control in Online Social Networks. In *Proceedings of the ACM Symposium on Access Control Models and Technologies* (Shanghai, China, June 6 - 8, 2016). SACMAT '16. ACM, New York, NY, 13-23. DOI= https://doi.org/10.1145/2914642.2914644.

[9] Hu, H., Ahn, G.-J. and Jorgensen, J. 2011. Detecting and Resolving Privacy Conflicts for Collaborative Data Sharing in Online Social Networks. In *Proceedings of the Annual Computer Security Applications Conference* (Orlando, FL, USA, December 5 - 9, 2011). ACSAC '11. ACM, New York, NY, 103-112. DOI= https://doi.org/10.1145/2076732.2076747.

[10] Hu, H., Ahn, G.-J. and Jorgensen, J. 2012. Enabling Collaborative Data Sharing in Google+. In *Proceedings of the IEEE Global Communications Conference* (Anaheim, CA, USA, December 3 - 7, 2012). GLOBECOM '12. IEEE, New York, NY, 720-725. DOI= https://doi.org/10.1109/glocom.2012.6503198.

[11] Palomar, E., Gonzalez-Manzano, L., Alcaide, A. and Galan, A. 2016. Implementing a Privacy-enhanced Attribute-based Credential System for Online Social Networks with Co-ownership Management. *IET Inform. Secur.*, 10, 2 (Mar 2016), 60-68. DOI= https://doi.org/10.1049/iet-ifs.2014.0466.

[12] Xu, K., Guo, Y., Guo, L., Fang, Y. and Li, X. 2015. My Privacy My Decision: Control of Photo Sharing on Online Social Networks. *IEEE Trans. Depend Secure*, PP, 99 (Jun. 2015), 1-13. DOI= https://doi.org/10.1109/tdsc.2015.2443795.

[13] Ilia, P., Polakis, I., Athanasopoulos, E., Maggi, F. and Ioannidis, S. 2015. Face/off: Preventing Privacy Leakage from Photos in Social Networks. In *Proceedings of the ACM Conference on Computer and Communications Security* (Denver, CO, USA, October 12 - 15, 2015). CCS '15. ACM, New York, NY, 781-792. DOI= https://doi.org/10.1145/2810103.2813603.

[14] Yuan, L., Korshunov, P. and Ebrahimi, T. 2015. Secure JPEG Scrambling Enabling Privacy in Photo Sharing. In *Proceedings of the IEEE International Conference and Workshops on Automatic Face and Gesture Recognition* (Washington, WA, USA, May 4 - 8, 2015). FG '15. IEEE, New York, NY, 1-6. DOI= https://doi.org/10.1109/fg.2015.7285022.

[15] Zhou, W., Li, H., Lu, Y. and Tian, Q. 2013. SIFT Match Verification by Geometric Coding for Large-Scale Partial-Duplicate Web Image Search. *ACM Trans. Multim. Comput.*, 9, 1 (Feb. 2013), 1-18. DOI= https://doi.org/10.1145/2422956.2422960.

[16] Taigman, Y., Yang, M., Ranzato, M. A. and Wolf, L. 2014. DeepFace: Closing the Gap to Human-Level Performance in Face Verification. In *Proceedings of the IEEE Conference on*

Computer Vision and Pattern Recognition (Columbus, OH, USA, June 24 - 27, 2014). CVPR '14. IEEE, New York, NY, 1701-1708. DOI= https://doi.org/10.1109/cvpr.2014.220.