Secure JPEG Scrambling enabling Privacy in Photo Sharing

Lin Yuan, Pavel Korshunov, and Touradj Ebrahimi
Multimedia Signal Processing Group, EPFL, Lausanne, Switzerland

***Abstract*— With the popularization of online social networks (OSNs) and smart mobile devices, photo sharing is becoming a part of people' daily life. An unprecedented number of photos are being uploaded and shared everyday through online social networks or photo hosting services, such as Facebook, Twitter, Instagram, and Flickr. However, such unrestrained online photo or multimedia sharing has raised serious privacy concerns, especially after reports of citizens surveillance by governmental agencies and scandalous leakage of private photos from prominent photo sharing sites or online cloud services. Popular OSNs typically offer privacy protection solutions only in response to the public demand and therefore are often rudimental, complex to use, and provide limited degree of control and protection. Most solutions allow users to control either who can access the shared photos or for how long they can be accessed. In contrast, in this paper, we take a structured privacy by design approach to the problem of online photo privacy protection. We propose a privacy-preserving photo sharing architecture based on a secure JPEG scrambling algorithm capable of protecting the privacy of multiple users involved in a photo. We demonstrate the proposed photo sharing architecture with a prototype application called ProShare that offers JPEG scrambling as the privacy protection tool for selected regions in a photo, secure access to the protected images, and secure photo sharing on Facebook.**

I. Introduction

Wide spread of smart mobile devices with high-resolution cameras and user-friendly social networks applications make photo sharing an easy and therefore popular activity. According to a survey conducted by Pew Research Center's Internet Project[1], more than half of internet users post or share photos and videos online and these numbers are rapidly growing. For instance, Instagram, which was launched about four years ago, already hosts more than 30 billion photos, with 70 million daily uploads on average[2].

However, most photo sharing services lack a sound solution for protecting users' privacy. Typically, social networks assume default public access for all information posted by a user, unless the user specifically restricts such access via a set of complicated privacy settings, making unaware users vulnerable and their privacy exposed. Many cloud-based photo storage services provide an easy free of charge photo sharing and management, but these services come at the cost of higher security risks, as shown by the recent scandal with private photos of celebrities leaked online[3]. Also, a large number of photo tags, caption information, and comments associated with online photos can be used to find and identify a person. Even if tags and comments do not explicitly identify a person, combined with face recognition and other publicly available data, they can be used to infer the identity with high accuracy [1]. Despite all these privacy risks, the the amount of photos posted online and shared on social networks is not declining. The majority of people are not fully aware of the potential privacy threats, while they enjoy advantages and conveniences of social networks.

In this paper, we explore and propose the design of a privacy-preserving photo sharing architecture, which ensures users privacy and at the same time preserves the usability and convenience of online photo sharing activity. Proposed architecture utilizes a multi-region selectively JPEG scrambling algorithm that ensures photo privacy for multiple people involved. Based on the proposed architecture, we built a prototype photo sharing application demonstrating the feasibility of the architecture and practicality of using secure JPEG scrambling for privacy protection.

The rest of the paper is structured as follows. Section II presents related work and motivation. Section III describes the secure JPEG scrambling. Section IV discusses the proposed privacy-preserving photo sharing architecture, which is based on the secure JPEG scrambling. Section V presents a photo sharing prototype iOS application, ProShare, which demonstrates the proposed architecture. Finally, Section VI concludes the paper and discusses possible future work.

II. Related Work

A lot of research efforts in image privacy have focused on approaches to incorporate privacy protection into existing security surveillance systems and frameworks, typically, by implementing access rights management and privacy policies [2][3][4]. Another large body of work is on the development of algorithms and methods to protect visual privacy, such as using watermarking to hide visual personal information [5], scrambling techniques to reversibly distort privacy sensitive regions [6], removal of unauthorized personnel from the video feed [7], and encoder independent geometrical-based reversible distortions [8][9].

Compared to video surveillance, online photo sharing has many different characteristics, e.g., photos shared in online social networks can be accessed and commented easily and quickly by many people with most photos being tagged with identification information. Therefore, online photo sharing applications demand a different and more integrated solution for privacy protection. An online photo sharing system should allow a secure and efficient way to recover protected information by people with correct access rights. It should

This work has been conducted in the framework of the Swiss SERI C12.0081, Eurostars ToFuTV, and EC funded Network of Excellence VideoSense.

[1]http://www.pewinternet.org/2013/10/28/photo-and-video-sharing-grow-online/

[2]http://instagram.com/press/

[3]http://www.mirror.co.uk/all-about/nude-celebrity-photos-leaked

also support a multiple-user functionality of online photo sharing to ensure protection of privacy for not only the uploader of a photo but also for others involved. Furthermore, context information including image metadata, photo caption, tags, user comments, etc., should be carefully treated. Besides, almost all photo sharing applications use JPEG as the image format, which calls for a special consideration of privacy protection in JPEG compressed images.

Various tools to ensure image privacy exist, including image filtering, encryption, and scrambling. Considering the fact that image filtering is usually non-reversible, conventional image filtering might not be a good choice. Since image data is characterized by a very high bitrate and a low commercial value compared with other types of data like banking data and confidential documents, conventional encryption techniques entail a significant complexity increase and are therefore not always optimal. Therefore, image scrambling or lightweight encryption can be considered instead as a secure and efficient tool to protect photo privacy.

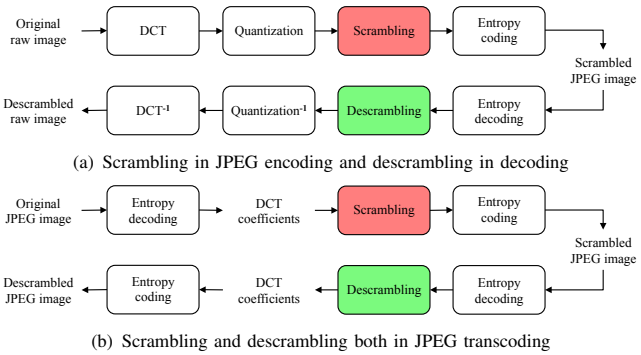
Many image scrambling techniques have been proposed by researchers. General image scrambling without the consideration of image coding usually works on pixel domain or bitstream directly, based on a chaotic map, e.g., Arnold scrambling [10], one-dimensional random scrambling [11], and other hybrid methods [12]. Scrambling in spatial domain has several disadvantages in its efficiency, complexity, and format compatibility. Taking into account the characteristics of JPEG data compression, scrambling in the bitstream or transform domain is more efficient. Most existing approaches to scrambling JPEG data are achieved by modifying its discrete cosine transformed (DCT) coefficients. Popular techniques include coefficient signs modification [13], cryptographic methods such as XOR operation [14], and coefficient permutation [15].

Moreover, a number of studies have been focused on privacy protection of photo and other media stored in social networks or cloud services. Researchers tried to understand users' privacy concern about photo sharing, as well as the potential privacy threats as subjective [16][17] and objective [18][19] studies show. Many approaches to privacy protection in photo sharing have been proposed, including usage control scheme in distributed OSNs [20], JPEG coding-based separate sharing [21], and tag-based access control [22][23]. However, most of these schemes have significant limitations in terms of security, efficiency, complexity, or usability. Therefore, more secure, efficient, and user friendly methods for insuring privacy of photo sharing need to be proposed.

III. Secure JPEG Scrambling

In this section, we describe in detail the secure JPEG scrambling algorithm, on which our privacy-preserving photo sharing algorithm is based.

We propose a multi-region selective JPEG scrambling scheme to protect visual privacy in photo for multiple users. This scrambling scheme is developed based on the secure JPEG framework by [13]. In such a scrambling scheme, one can scramble multiple regions of interest (ROIs) with



(a) Scrambling in JPEG encoding and descrambling in decoding

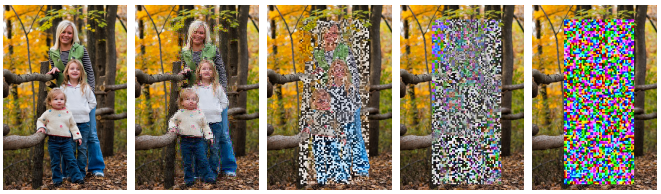
(b) Scrambling and descrambling both in JPEG transcoding

Fig. 1. Two modes of JPEG scrambling and descrambling.

arbitrary shapes in an images, using one or different secret keys. Each scrambled region is assigned with an ID and the descrambler can selectively descramble the regions using corresponding scrambling keys. The scrambling of the JPEG data is achieved by modifying the signs of the quantized discrete cosine transform (DCT) coefficients corresponding to the defined ROIs. Scrambling and descrambling processes can be done in not only JPEG encoding and decoding respectively, but JPEG transcoding. Scrambling a JPEG image in transcoding mode ensures a lossless reconstruction of the original image because transcoding process does not involve quantization and re-quantization operations and therefore does not affect the DCT coefficients outside the scrambled regions. Scrambling and descrambling in JPEG encoding/decoding and transcoding are illustrated by the diagrams shown in Fig. 1. The scrambling process can be described in detail as follows:

a) Region selection and key preparation: First of all, one can select several regions in an image, by providing a *mask matrix* $\mathbf{M}$, non-zero elements of which indicate the 16×16 Minimum Coded Unit (MCU) blocks of the image to be scrambled. The scrambled regions are restricted to match the MCU blocks boundaries. In this matrix, except for zero-valued elements, the value n of non-zero element is referred to as the ID of each scrambled region. We note each scrambled region as ROI_n. For each ROI_n, a secret scrambling key $\mathbf{k}_n$ and a scrambling strength level $l_n \in \{1, 2, 3, 4\}$ are defined. The scrambling key can be any value or sequence set by user. The scrambling strength is subdivided into four levels: 1-low, 2-medium, 3-high, and 4-ultra-high, meaning of which will be discussed later. Therefore, a mask matrix $\mathbf{M}$ with regions ID n defined, a set of secret scrambling keys $\mathbf{k}_n (n = 1, 2, 3, \ldots)$, and a corresponding set of scrambling strength levels $l_n (n = 1, 2, 3, \ldots)$ constitute the parameters to scramble an image.

b) DCT coefficients manipulation: In this step, the DCT coefficients corresponding to the defined scrambled regions are modified, according to the scrambling key and strength level. We note $x_i (i = 1, 2, 3, \ldots, 64)$ as the value of quantized DCT coefficients within an 8×8 DCT block. A pseudo random number generator (PRNG) initialized by a seed value is used to drive the scrambling process, where we simply use the scrambling key as the seed. The PRNG



(a) Original (b) Low (c) Medium (d) High (e) Ultra-high

Fig. 2. Scrambled images with different scrambling strength levels. (a) Original image; (b) Low-level scrambling; (c) Medium-level scrambling; (d) High-level scrambling; (e) Ultra-high-level scrambling. Example image is from the Images of Groups Dataset [24].

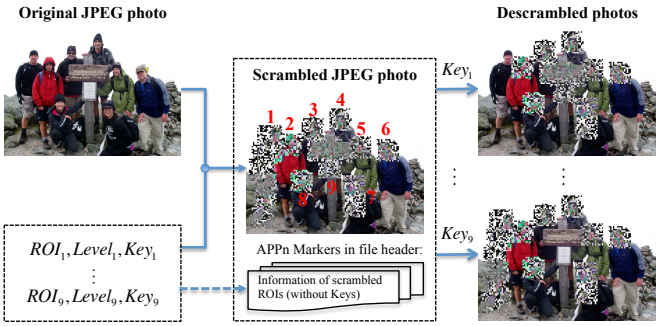


Fig. 3. Multi-region scrambling and selective descrambling. Example image is from The Images of Groups Dataset [24].

generates a random sequence of 1 and -1, which is multiplied with the DCT coefficients x_i. For low-level scrambling, only AC coefficients of all YUV components are modified; for medium-level scrambling, both DC and AC coefficients of only luminance (Y) component are modified; while for high-level scrambling, both DC and AC coefficients of all YUV components are changed. In case a stronger scrambling is needed (ultra-high-level), we can further scramble the DC coefficients, by performing a bitwise XOR operation between each DC value and a pseudo-random number with the same length in bits as the DC coefficient. Fig. 2 illustrates scrambled versions of an image with different strength levels.

c) Scrambling information insertion: Once an image is scrambled, information about the scrambled regions is inserted in one or more application markers (APPn) in JPEG file header. The information to be inserted includes the elements of the mask matrix $\mathbf{M}$ and the scrambling strength l_n, former of which records the location, shape and IDs of the scrambled regions. Therefore, the scrambled image is JPEG-compliant and can be viewed by a typical JPEG decoder. However, to descramble and view the original image, a special descrambler (decoder or transcoder) and the correct scrambling key(s) are needed.

Intuitively, descrambling process simply reverses the scrambling processes described above. Given a region ID, the descrambler can extract corresponding scrambling strength and region location and shape from APPn markers of JPEG header. As long as a correct scrambling key is provided, the region can be recovered. We have implemented the proposed scrambling algorithm in both transcoder and encoder/decoder using an open source JPEG library by Independent JPEG Group[4]. The process of a multi-region scrambling and selective descrambling is illustrated in Fig. 3.

Furthermore, not only visual information is privacy sensitive, but metadata associated with a photo can also reveal personal information, e.g., geo-location data and the time when the photo was captured. It is therefore important to protect the privacy information in metadata as well. Metadata of a JPEG photo is recorded with an Exchangeable image file format (Exif) tag, stored in APP1 marker of JPEG header. Several approaches to privacy protection of metadata exist, including hiding Exif information in JPEG DCT coefficients [25] or simply removing all metadata. However, neither of the approaches meet the privacy and utility requirements of a photo sharing system. To conceal the metadata and also ensure its reuse, we propose encrypting selected JPEG metadata in the Exif tag.

Experiment

Since scrambling of a JPEG image modifies the DCT coefficients and introduces more information in the JPEG file header, the file size of the scrambled image can be increased when compared to the original image. Besides, how well each scrambling strength level can protect photo privacy using such a scrambling algorithm is not very clear. Therefore, we conduct an experiment to examine the influence of scrambling on the size of overhead and performance of automatic face detection. We use Haar face detector from the OpenCV library[5] and apply it on 1000 images from the Images of Groups Dataset [24]. For each image, we scramble the detected faces regions, and scramble the whole image respectively, using four levels of scrambling strength, and then calculate the increased files sizes (in percentage) of the two kinds of scrambled images (face region scrambled and whole image scrambled) when compared to original images. Finally, face detection is again applied on the face scrambled images and the total number of detected faces are recorded. Experimental results are shown in Table I. According to the results, we first observe that the scrambling significantly reduces the number of detected faces especially for the scrambling strength above low level. Actually, if we look into the small number of detected faces in the scrambled images (medium, high and ultra-high levels), most of them are wrongly detected due to the blocking artifacts created by scrambling. Second, such a scrambling algorithm introduces very low bitrate overhead, especially for the scrambling strength below ultra-high level. As low to high level scrambling changes only the signs of DCT coefficients, size of the scrambled image does not increase much even if the whole image is scrambled. However, XOR operation employed in the ultra-high level scrambling can drastically change the DC value of DCT coefficients and therefore impacts more the entropy coding and introduces larger overhead, especially for

[4]IJG: http://www.ijg.org/

[5]Open source computer vision: http://opencv.org/

TABLE I

IMPACT OF SCRAMBLING ON BITRATE OVERHEAD AND PERFORMANCE OF AUTOMATIC FACE DETECTION.

	Original image	Low-level scrambled	Medium-level scrambled	High-level scrambled	Ultra-high-level scrambled
Total number of detected faces	3944	1638	14	11	10
Average overhead (face region scrambled)	–	1.87%	2.04%	2.15%	3.15%
Average overhead (whole image scrambled)	–	1.87%	4.89%	5.96%	18.41%

large scrambled areas. Therefore, a medium or high level is considered to be a preferable scrambling strength to achieve a trade-off between privacy and bitrate overhead. However, this should be verified through further evaluations including automatic face recognition and subjective evaluations.

IV. PRIVACY-PRESERVING PHOTO SHARING

In this section, we describe the design of a privacy-preserving photo sharing architecture that is based on the secure JPEG scrambling.

Architecture Overview

The architecture consists of two key parts: (i) a client-side application for securing photos and (ii) a private server for hosting photos and managing users accounts. We consider that all local client-side components (operating system, applications, sensors, etc.) are trustworthy, while the server is trustworthy only with some reservations and under certain conditions, which are discussed in Section IV-B. This is based on the assumption that social networks are often considered as untrusted. Fig. 4 presents the proposed architecture in this paper.

In our architecture, a client application can apply the JPEG scrambling to a photo using one or more secret keys. The photo is then uploaded to the dedicated server, which is designed to host only protected photos. Other users within this architecture can view the photo by requesting and downloading the photo and then "unlock" the photo with a key. In case of sharing this photo on a public social network, the server acts as a "bridge" between the photo sender and the social network. Only a link to the secure photo will be posted to the social network along with an eventual protected image or its thumbnail, so that authorized viewers are bound to use a secure server to access the photo. A correct secret key is required for viewing the corresponding original photo. The sharing process can therefore be split into three tasks or operations: sender-side, server-side, and recipient-side operations.

A. Sender-side Operation

When a sender attempts to transmit a photo taken by a personal device's built-in camera or selected from a photo album, client-side application provides options for the sender to protect the photo, with several alternative Secure JPEG scrambling tools described in Section III. In this process, privacy information of the photo is protected by scrambling using a secret key (or a set of keys) set by the sender. To make sharing multiple photos for the sender easier, face and

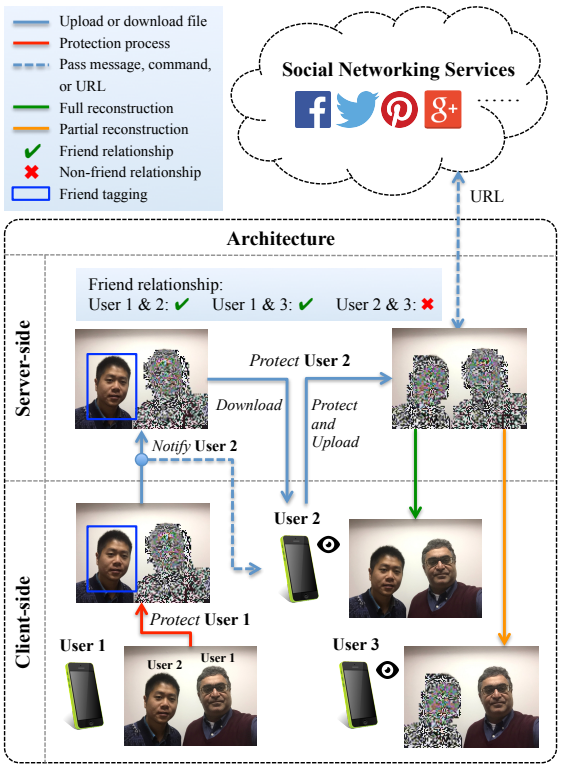


Fig. 4. Proposed privacy-preserving photo sharing architecture.

object detection algorithms can be applied on the images to identify privacy sensitive regions. In the case where there are several people in a photo, e.g., friends or family of the sender, the sender can either protect his friends using the same or a different key, or tag their friends to let them decide how to best protect their own privacy using protection keys defined by themselves.

B. Server-side Operation

The server is designed to act as a normal photo sharing service like Facebook or Instagram. However, this server hosts only secure photos uploaded by users, which is the most important feature of the proposed architecture. Decoding and display of original photos happen in the client-side. Besides photo hosting, the server has also a simple user account and access management system similar to other social networks. In the current design of the architecture, friendship can have a hierarchical structure, for instance, one can categorize his friends into different groups: *intimate*, *normal*, and *unfamiliar*. Alternatively, the server can utilize existing friendship relations from one of the social networks.

For a friend in different groups, one can selectively expose protected regions of a photo, by sharing different protection keys corresponding to different protection regions or objects. A Public Key Infrastructure (PKI) can be used to distribute secret keys between the sender and one or more recipients. All image transformation (scaling, cropping, filtering, etc.) is performed at the client side prior to uploading of an image and the server does not apply any further processing as it often happens in many photo-sharing and social network services. Since secure JPEG scrambling can be lossless, the server can be viewed as a high-quality image hosting system with privacy enhancement features.

C. Recipient-side Operation

There are two ways for a recipient to view a photo: (i) via a client application on the device, and (ii) via a URL posted on social networks. In the former case, a recipient who is authorized by a photo sender can download a secure photo from the server, reconstruct a clear version, and view the photo in a client device, as how *User 2* and *User 3* view a photo uploaded by *User 1* in Fig. 4. A special JPEG decoder/transcoder with descrambling is incorporated in the client application to ensure the reconstruction of the original unprotected version from a secure photo on the client device. In the latter case, a photo sender can share the photo on Facebook, by posting an external link to the secure photo in the server. Authorized Facebook users who attempt to view the photo will be directed to the server. Only those who explicitly or implicitly possess the secret key(s) can reconstruct the photo partially or completely. However, a user who wants to view the photo using a web browser without a secure JPEG scrambling plugin will have to rely on the server to perform descrambling. So the current design of the proposed architecture relies on the server to reconstruct a photo temporarily for display. In this case, the completely or partically reconstructed photo is exposed temporarily to the server, which is why we assume the server to be conditionally trustworthy. This issue can be solved by using a local HTTP/HTTPS proxy, similar to the approach proposed in [21]. Access to a secure photo goes through the local proxy and reconstruction of the photo is performed by the local proxy. However, in the future, the adoption of a Secure JPEG scrambling standard or a widely used plugin will allow a client web browser to also reconstruct (decode) a secure JPEG scrambled photo directly without the temporal exposure of the unprotected image to the server.

Discussion

As mentioned in Section IV-B, we assume the existence of a Public Key Infrastructure (PKI) to manage the secret key distribution between a sender and one or more recipients and to authorize the complete or partial reconstruction of a secure photo by the recipient. The exchange of the secret keys can be based on a public key cryptography. Identity verification of the recipient, when using such PKI, can rely on an access management system of the server or on the information about the friendship obtained from a social network, where the photo is being shared. Implementation of PKI raises several issues that need to be addressed in a practical implementation, including flexibility, security, scalability, and trustworthiness. However, PKI and secret key management is out of the scope of this paper, since the main focus is on privacy-protecting photo sharing based on Secure JPEG scrambling.

The proposed architecture design has a significant impact on photo privacy protection in online social networks and related applications. By applying Secure JPEG scrambling, both privacy sensitive visual information and metadata of a photo are protected or distorted, making any analysis and retrieval of such information harder. Therefore, this design effectively reduces three kinds of privacy-related threats currently present in online photo sharing ecosystem: (i) unauthorized access to photos, (ii) automatic identification and recognition, and (iii) image data mining. Last but not least using Secure JPEG scrambling makes minimal impact on current system workflow, bandwidth usage, and data storage, since images protected with Secure JPEG scrambling have similar bitrates with minimal overhead when compared to typical JPEG images.

V. Prototype Application

A prototype application called ProShare was built to demonstrate the proposed photo sharing architecture. Fig. 5 shows several screenshots of the ProShare application. The prototype consists of two components: (i) a client iOS platform application and (ii) a server for image storage and processing. The prototype uses secure JPEG scrambling as the protection tool to protect photo privacy. The application performs scrambling on a photo according to the region and scrambling strength set by the user. Only the scrambled

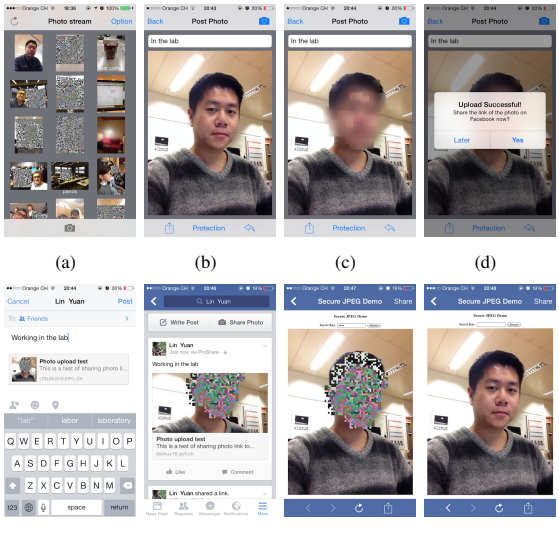


Fig. 5. Example screenshots of ProShare, tested on iPhone 5C. (a) photo stream; (b) take a new photo; (c) define a scrambling region by finger touching; (d) upload the photo to private server successfully; (e) posting link of the photo to Facebook; (f) posted link on Facebook; (g) web page showing the scrambled photo; (h) web page showing the descrambled photo, when correct key provided.

photo is stored on the server. Using Facebook iOS API, the application allows the user to share the secure photo on Facebook along with a URL pointing to the server. By following the link, other Facebook users would only see the scrambled photo unless they can provide a secret key, in which case a descrambled (clear) photo is shown. Within the ProShare application, multiple users can upload their photos to the server and everyone can see the scrambled photos uploaded by other users in a *Photo Stream* page (Fig. 5(a)). Only own user's photos are descrambled automatically. For photos of other users the correct secret key is necessary.

Since ProShare application is still under development, some features have not been fully implemented, e.g., scrambling and descrambling on client application and automatic key distribution. Currently, for the ease of implementation and demonstration purposes, the photo is protected on the server directly and a simple key verification scheme is implemented. Nevertheless, such a simple prototype can already validate the proposed photo sharing architecture in many practical use cases. For instance, the application can be used to hide personal information (name, address, date of birth, etc.) on sensitive documents, such as banking statements, IDs, passports and airplane tickets.

VI. Conclusion

In this paper, we describe a secure JPEG scrambling scheme, which ensures the protection of visual information of multiple regions in an image as well as photo metadata. The protected regions can be selectively descrambled depending on the region ID and scrambling key given by user. An architecture is proposed for privacy-preserving photo sharing. Such an architecture keeps only secure photos in online servers, while the protection, reconstruction, and viewing of photos are performed on the client devices. To demonstrate the proposed architecture, we built a prototype iOS application ProShare that enables privacy protection of photos shared online. Although still under development, the prototype application shows a good degree of usability of proposed photo sharing architecture. Future work lies in the further evaluation of the proposed photo sharing architecture and prototype application.

References

[1] A. Acquisti, R. Gross, and F. Stutzman. (2011, August) Faces of facebook: Privacy in the age of augmented reality. A YouTube presentation at Blackhat USA Technical Security Conference. [Online]. Available: http://www.truststc.org/pubs/834.html

[2] T. Ebrahimi, Y. Abdeljaoued, R. F. I. Ventura, and O. D. Escoda, "MPEG-7 camera," in *Proceedings of International Conference on Image Processing*, vol. 3, 2001, pp. 600–603.

[3] A. Senior, S. Pankanti, A. Hampapur, L. Brown, Y.-L. Tian, A. Ekin, J. Connell, C. Shu, and M. Lu, "Enabling video privacy through computer vision," *IEEE Security and Privacy*, vol. 3, no. 3, pp. 50–57, May 2005.

[4] A. J. Aved and K. A. Hua, "A general framework for managing and processing live video data with privacy protection," *Multimedia Syst.*, vol. 18, no. 2, pp. 123–143, 2012.

[5] W. Zhang, S. Cheung, and M. Chen, "Hiding privacy information in video surveillance system," in *Proc. IEEE International Conference on Image Processing*, Genoa, Italy, Sep 2005.

[6] F. Dufaux and T. Ebrahimi, "Scrambling for privacy protection in video surveillance systems," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 18, no. 8, pp. 1168–1174, Aug 2008.

[7] J. Wickramasuriya, M. Datt, S. Mehrotra, and N. Venkatasubramanian, "Privacy protecting data collection in media spaces," in *Proceedings of the 12th annual ACM international conference on Multimedia*, New York, NY, USA, Oct. 2004, pp. 48–55.

[8] P. Korshunov and T. Ebrahimi, "Using warping for privacy protection in video surveillance," in *18th International Conference on Digital Signal Processing (DSP)*, Santorini, Greece, 2013, pp. 1–6.

[9] ——, "Using face morphing to protect privacy," in *IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, Krakow, Poland, Aug. 2013, pp. 208–213.

[10] Z. Tang and X. Zhang, "Secure image encryption without size limitation using Arnold transform and random strategies." *Journal of Multimedia*, vol. 6, no. 2, pp. 202–206, 2011.

[11] Q. Sun, P. Guan, Y. Qiu, and Y. Xue, "A novel digital image encryption method based on one-dimensional random scrambling." in *FSKD*. IEEE, 2012, pp. 1669–1672.

[12] L. Zhang, X. Tian, and S. Xia, "A scrambling algorithm of image encryption based on Rubik's cube rotation and logistic sequence," in *Multimedia and Signal Processing (CMSP), 2011 International Conference on*, vol. 1, May 2011, pp. 312–315.

[13] F. Dufaux and T. Ebrahimi, "Toward a Secure JPEG," in *Proc. SPIE*, vol. 6312, 2006, pp. 63 120K–63 120K–8.

[14] T. Honda, Y. Murakami, Y. Yanagihara, T. Kumaki, and T. Fujino, "Hierarchical image-scrambling method with scramble-level controllability for privacy protection," in *2013 IEEE 56th International Midwest Symposium on Circuits and Systems*, Aug 2013, pp. 1371–1374.

[15] K. Wong and K. Tanaka, "DCT based scalable scrambling method with reversible data hiding functionality," in *2010 4th International Symposium on Communications, Control and Signal Processing*, March 2010, pp. 1–4.

[16] S. Ahern, D. Eckles, N. Good, S. King, M. Naaman, and R. Nair, "Over-exposed?: Privacy patterns and considerations in online and mobile photo sharing." in *CHI*, M. B. Rosson and D. J. Gilmore, Eds. ACM, 2007, pp. 357–366.

[17] A. Besmer and H. Richter Lipford, "Moving beyond untagging: Photo privacy in a tagged world," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '10. New York, NY, USA: ACM, 2010, pp. 1563–1572.

[18] G. Friedland and R. Sommer, "Cybercasing the joint: On the privacy implications of geo-tagging," in *Proceedings of the 5th USENIX Conference on Hot Topics in Security*, ser. HotSec'10. Berkeley, CA, USA: USENIX Association, 2010, pp. 1–8.

[19] J. P. Pesce, D. L. Casas, G. Rauber, and V. Almeida, "Privacy attacks in social media using photo tagging networks: A case study with Facebook," in *Proceedings of the 1st Workshop on Privacy and Security in Online Social Media*, ser. PSOSM '12. New York, NY, USA: ACM, 2012, pp. 4:1–4:8.

[20] L. A. Cutillo, R. Molva, and M. Önen, "Privacy preserving picture sharing: Enforcing usage control in distributed on-line social networks," in *SNS 2012, 5th ACM Workshop on Social Network Systems*, Bern, Switzerland, April 2012.

[21] M.-R. Ra, R. Govindan, and A. Ortega, "P3: Toward privacy-preserving photo sharing," in *Presented as part of the 10th USENIX Symposium on Networked Systems Design and Implementation*. Berkeley, CA: USENIX, 2013, pp. 515–528.

[22] P. Klemperer, Y. Liang, M. Mazurek, M. Sleeper, B. Ur, L. Bauer, L. F. Cranor, N. Gupta, and M. Reiter, "Tag, you can see it!: Using tags for access control in photo sharing," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '12. New York, NY, USA: ACM, 2012, pp. 377–386.

[23] M. L. Mazurek, Y. Liang, W. Melicher, M. Sleeper, L. Bauer, G. R. Ganger, N. Gupta, and M. K. Reiter, "Toward strong, usable access control for shared distributed data," in *Proceedings of the 12th USENIX Conference on File and Storage Technologies (FAST 14)*. Santa Clara, CA: USENIX, 2014, pp. 89–103.

[24] A. Gallagher and T. Chen, "Understanding images of groups of people," in *Proc. CVPR*, 2009.

[25] M. Niimi, F. Masutani, and H. Noda, "Protection of privacy in JPEG files using reversible information hiding," in *2012 International Symposium on Intelligent Signal Processing and Communications Systems*, Nov 2012, pp. 441–446.