Privacy-Preserving Photo Sharing based on a Secure JPEG

Lin Yuan, Pavel Korshunov, and Touradj Ebrahimi
Multimedia Signal Processing Group, EPFL, Lausanne, Switzerland
Email: {lin.yuan, pavel.korshunov, touradj.ebrahimi}@epfl.ch

***Abstract*—Sharing photos online is a common activity on social networks and photo hosting platforms, such as Facebook, Pinterest, Instagram, or Flickr. However, after reports of citizens surveillance by governmental agencies and the scandalous leakage of celebrities private photos online, people have become concerned about their online privacy and are looking for ways to protect it. Popular social networks 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 either allow users to control 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 that takes into account content and context of a photo with privacy protection integrated inside the JPEG file itself in a secure way. We demonstrate the proposed architecture with a prototype mobile iOS application called ProShare that offers scrambling as the privacy protection tool for a selected region in a photo, secure access to the protected images, and secure photo sharing on Facebook.**

I. Introduction

As we are stepping into the so called *Big Data* era, various types of personal information are being collected in unprecedented ways. Not only conventional personal information, such as name, age, and birthday, but also personal photos, geo-locations, personal interests, lists of friends, and activity records are collected and mined by online social networks. Most of the time, people share such information voluntarily for personal convenience, for entertainment reasons, or to simply "show off" to others.

Wide spread of smart mobile devices with high-resolution cameras and user-friendly social networks applications makes 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 scheme 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, use of online photo sharing does not seem to diminish. Majority of people lack awareness of potential privacy threats while enjoying the advantages and conveniences brought by social networks and big data.

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 Secure JPEG framework that integrates different tools to protect photo privacy. Based on the proposed architecture, we built a prototype photo sharing application demonstrating the feasibility of the architecture and privacy protection tools.

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

II. Background and Motivation

A lot of research efforts on image privacy in the past were focused on approaches to incorporate privacy protection into existing security surveillance systems and frameworks, typically via implementing access rights management and policies [2][3][4]. Another large body of work is on 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], encoder independent geometrical-based reversible distortions [8][9].

[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

(a) Original (b) Masking (c) Pixelation (d) Inpainting (e) Scrambling

Fig. 1. Different visual privacy protection tools for Secure JPEG framework.

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 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], coefficient permutation [15], etc. In [13], Dufaux and Ebrahimi propose a Secure JPEG, an open and flexible framework to secure JPEG images, allowing for efficient integration and use of different security tools. With a Secure JPEG, various image processing techniques including image filtering, masking, inpainting, morphing and also encryption and scrambling can also be applied in a reversible way. Another advantage of using a Secure JPEG is that it does not affect syntax compliance of processed JPEG data. Further discussion about privacy protection using a Secure JPEG will be made in Section III.

Despite a large number of research works on image security, encryption or scrambling techniques, research effort specially on privacy protection in online photo sharing is still insufficient. Some researchers tried to understand users' privacy concern on photo sharing as well as the potential privacy threads, through subjective [16][17] and objective [18][19] studies. Many approaches towards privacy protection in photo sharing have been proposed, including usage control scheme in distributed OSNs [20], separately sharing secret and public parts of image based on JPEG [21] and tag-based access control [22][23]. However, most approaches have significant limitations in their security, efficiency, complexity or usability. Therefore, more secure, efficient and usable approaches to insure privacy in photo sharing need to be explored.

III. Secure JPEG Framework

Before introducing the proposed architecture for privacy protection of online photo sharing, we present a Secure JPEG framework, the technology and image file format, on which our architecture is based.

Secure JPEG is an open, flexible, reversible and format compliant framework to secure JPEG images, which allows an efficient integration and use of different security and privacy protection tools (see examples in Fig. 1). Information about the original pixels and protected metadata, along with the protection parameters are securely hidden in one or more JPEG application markers. Protection and reconstruction relies on a or more secret keys, which are transmitted separately in a secure way between a sender (a person who shares a photo) and trusted recipients (trusted people to share photo with). Multiple regions can be protected with different keys to enable hierarchical privacy protection.

Visual privacy in a JPEG photo can be protected using different visual distortions classified into two main types:

- *Pixel replacement*: these techniques replace the pixels of an original image with other masks, distortions, or patterns. Simple blurring, pixelation, masking, or more complex methods such as inpainting can be used to replace the original pixel regions. The original pixels are then compressed, encrypted, and embedded, together with information about position and shape of the processed regions, via one or more APPn markers inside JPEG header. Reconstruction of original image is performed by extracting from JPEG header, decrypting, and placing back the original pixels.

(a) Original (b) Low-level (c) Medium-level (d) High-level

Fig. 2. Scrambled images with different scrambling levels. (b) Low-level: scramble only AC coefficients for all YUV components; (c) Medium-level: scramble both DC and AC coefficients for only luminance (Y) component; (d) High-level: scramble both DC and AC coefficients for all YUV components. Example image is from The Images of Groups Dataset [24].

- *Data manipulation*: these techniques do not replace the original pixels but change them in a specific way. Typical examples include image encryption and scrambling, which modify the original values of image pixels, DCT coefficients, or bitstream in a reversible way with the help of a secret key. Only information about the shape of the protected regions needs to be embedded in APPn markers. Therefore, these methods introduce less overhead to the image bitrate compared to pixel replacement methods. Only a correct secret key is needed for the reconstruction of the original image.

Each of the above types has advantages and drawbacks. Pixel replacement can be robust to such image processing like cropping, resizing, and filtering, which are often performed on images by social networks. As long as these image processing operations are known, an image can be reconstructed. Pixel replacement techniques can also create interesting and visually attractive visual effects, e.g., inpainting or masking with a smiley face, which can be an important factor in adoption of such approaches by the users. However, these techniques can significantly increase overhead to file size. Data manipulation methods, on the other hand, have little impact on the file sizes and are less complex, because the original image data does not need to be stored. The secret key is the most important information for securing and also recovering an image. However, these methods are less robust to image processing operations. Any modification on the secured region of the image can potentially make the protection irreversible.

Not only is visual information privacy sensitive, but also metadata associated with a photo can also reveal personal information, e.g., geo-location data and the time when the photo is created. 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 meets 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.

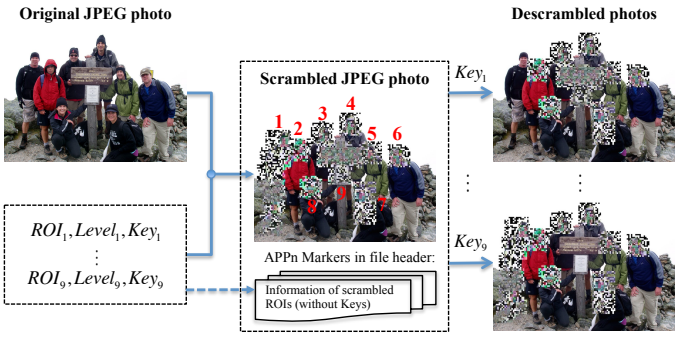

Fig. 3. Multi-region scrambling (high-level) and selective descrambling. Example image is from The Images of Groups Dataset [24].

Multi-Region Selective JPEG Scrambling

We illustrate a typical image privacy protection scenario using a scrambling technique. In a practical scenario, it is also important to ensure an independent secure protection of each region in a given image. For this reason, we developed a multi-region selective JPEG scrambling scheme. In this scheme, each scrambled region is assigned with an ID, so that the descrambler can selectively descramble the regions. Scrambling of JPEG data is achieved by flipping the signs of the quantized DCT coefficients in a random way, by using a pseudo-random algorithm initialized with a seed value. The seed value can be set by the users and is treated as a secret key (also called a scrambling key) for a region. Fig. 2 shows an example of scrambled images with different scrambling strengths. Once each region is scrambled, information about the location, shape and scrambling level of each region is inserted in one or more APPn markers. Therefore, the scrambled image is JPEG-compliant and can be viewed by typical JPEG viewers. However, to view the original image, a special descrambler (decoder or transcoder) is needed to descramble the image. Descrambling process simply reverses the scrambling processes described above. Given a region ID, descrambler can extract corresponding information about the scrambled region from APPn markers in JPEG header. As long as a correct scrambling key is provided, the region can be recovered. A multi-region scrambling and selective descrambling is illustrated by Fig. 3. We have implemented presented scrambling algorithm in both transcoder and encoder/decoder based on open source JPEG library by IJG[4]. Scrambling in JPEG transcoding can ensure lossless reconstruction.

IV. Architecture

In this section, we describe in detail the design of a privacy-preserving photo sharing architecture that is based on the Secure JPEG framework.

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

[4]Independent JPEG Group: http://www.ijg.org/

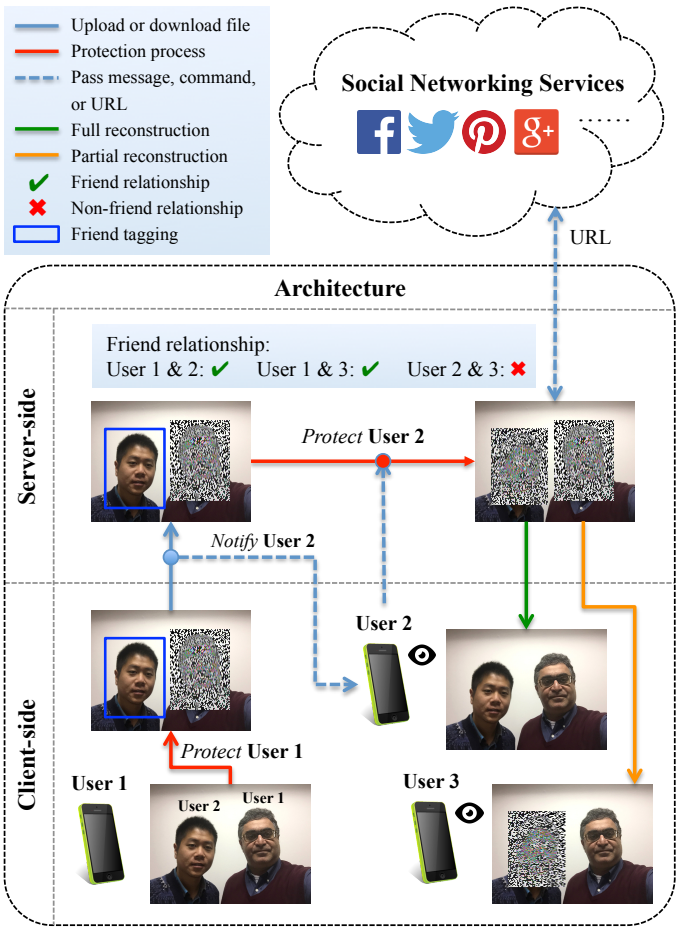

Fig. 4. Proposed privacy-preserving photo sharing architecture.

trustworthy only with some reservations and under certain conditions, which are discussed in Section IV-C. The photo storage and hosting systems of public social networks are assumed to be untrusted, while their friendship management can be trusted. An example of the proposed architecture is presented in Fig. 4.

In such an architecture, a client application can apply Secure JPEG protection 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 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 tools described in Section III. In this process, privacy information of the photo is protected by using the selected protection tool and a secret key (or a set of keys) set by the sender. To make sharing multiple photos for the sender easier, face and object detection algorithms can be applied on the images to identify privacy sensitive regions. A photo with several people in it, e.g., friends or family of the sender, raises various privacy issues, which can be described by the following scenarios:

- *Scenario 1:* A typical scenario where the sender treats the privacy of all individuals in the photo as his own and protects their images with the same or several corresponding secret keys. This scenario can be used when those individuals are not members of the photo sharing system and when they cannot be identified.
- *Scenario 2:* A common scenario where the photo is taken with close friends and other identifiable individuals and the sender wants to inform them and receive their approval. In this case, the sender can protect his friends using different keys and tag these friends. After the photo is uploaded on the server, each friend will be notified and can confirm the method of protection applied by the sender.
- *Scenario 3:* A similar scenario to *Scenario 2* where the sender is not sure about how best protect friends privacy and provides them with a greater degree of control over their own privacy. In this case, the sender will only tag friends in the photo. Once these friends get notification from tags, they can decide how to best protect their own privacy with protection keys defined by themselves. This scenario is illustrated by Fig. 4.
- *Scenario 4:* A scenario where the sender takes no privacy protection measures. This last scenario, although possible, is discouraged by the proposed architecture and solutions discussed in this paper.

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 friendship management system similar to other social network services. 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 cryptography 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 a Secure JPEG protection 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 unprotected 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 Secure JPEG decoder 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 plugin will have to rely on the server to perform image reconstruction. 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 partially 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 standard or a widely used plugin will allow a client web browser to also reconstruct (decode) a Secure JPEG photo directly without the temporal exposure of the unprotected image to the server.

Discussion

1) Secret key distribution: As mentioned in Section IV-B, we assume the existence of a public-key cryptography 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 public keys can be based on a public key infrastructure (PKI). Conventional PKI models based on either certification authorities (CA) or decentralized webs-of-trust have not become very popular due to their disadvantages in flexibility, security, scalability, and trustworthiness. Therefore a social network based PKI [26][27][28] can be employed for identity verification of the recipients, relying on a friendship management system of the server or on the information about the friendship obtained from a social network, where the photo is being shared. Also, this requires the friendship management system of the server or social networks to be trustworthy. However, PKI and secret key management is out of the scope of this paper, since the main focus is on privacy-protecting photo sharing architecture.

2) Significance to privacy protection in Big Data: The proposed architecture design has a significant impact on Big Data and related applications. By applying Secure JPEG, 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 existing in online photo sharing ecosystem: (i) unauthorized access to photos, (ii) automatic identification recognition, and (iii) image data mining. Last but not least using Secure JPEG makes minimal impact on current system workflow, bandwidth usage, and data storage, since images protected with Secure JPEG have similar bitrate compared to typical JPEG images.

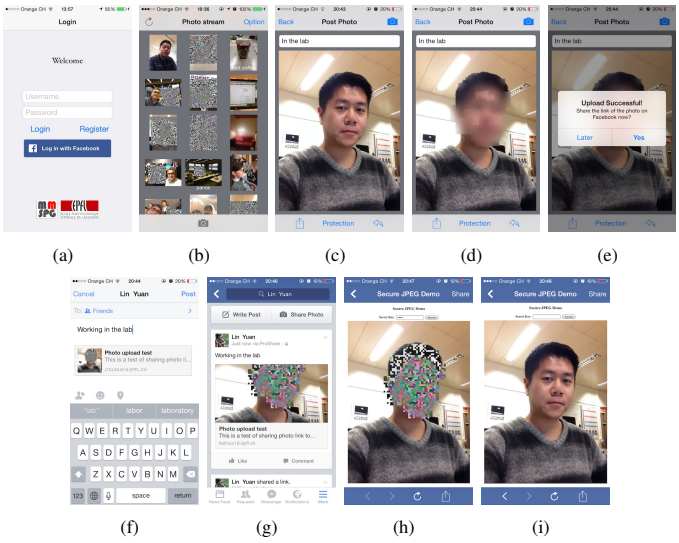

Fig. 5. Example screenshots of ProShare, tested on iPhone 5C. (a) login page of ProShare iOS application; (b) photo stream; (c) take a new photo; (d) define a scrambling region by finger touching; (e) upload the photo to private server successfully; (f) posting link of the photo to Facebook; (g) posted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

V. Prototype Application

A prototype application named *ProShare* was built to demonstrate the proposed photo sharing architecture. Fig. 5 shows several screenshots of 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 scrambling as the protection tool for Secure JPEG. The application performs scrambling on a photo according to the region and scrambling level set by the user. Only the scrambled secure 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 scrambled photo unless they can provide a secret key, in which case a descrambled (original) photo is shown. Within the ProShare iOS 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. 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 utility bills, banking statements, IDs, passports, airplane tickets, and so on.

VI. Conclusion

In this paper, we propose a privacy-preserving photo sharing architecture based on Secure JPEG, an open and flexible framework for ensuring photo privacy using various security and privacy protection tools. The 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 the photos shared online. Although still under development, the prototype application shows a good degree of usability of the proposed photo sharing architecture. Future work lies in further evaluation of privacy and user experience of the proposed photo sharing architecture and application.

Acknowledgment

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

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.

[26] V. Gruhn, M. Hülder, and V. Wolff-Marting, "Utilizing social networking platforms to support public key infrastructures," in *Proceedings of the Second International Conference on Security and Cryptography (SECRYPT 2007)*. INSTICC, 2007, pp. 245–250.

[27] R. Baden, A. Bender, N. Spring, B. Bhattacharjee, and D. Starin, "Persona: An online social network with user-defined privacy," *SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 4, pp. 135–146, Aug. 2009.

[28] V. Narayanan, G. Rose, and L. Dondeti, "Social network based PKI authentication," Nov. 2012, US Patent App. 13/419,065. [Online]. Available: http://www.google.com/patents/US20120278625