Processing-Aware Privacy-Preserving Photo Sharing over Online Social Networks

Weiwei Sun
Department of CIS
University of Macau
yb57457@umac.mo

Jiantao Zhou
Department of CIS
University of Macau
jtzhou@umac.mo

Ran Lyu
Department of CIS
University of Macau
mb45442@umac. mo

Shuyuan Zhu
School of EE
UESTC
eezsy@uestc.edu.cn

ABSTRACT

With the ever-increasing popularity of mobile devices and online social networks (OSNs), sharing photos online has become extremely easy and popular. The privacy issues of shared photos and the associated protection schemes have received significant attention in recent years. In this work, we address the problem of designing privacy-preserving, high-fidelity, storage-efficient photo sharing solution over Facebook. We first conduct an in-depth study on the manipulations that Facebook performs to the uploaded images. With the awareness of such information, we suggest a DCT-domain image encryption scheme that is robust against these lossy operations. As validated by our experimental results, superior performance in terms of security, quality of the reconstructed images, and storage cost can be achieved.

Keywords

Social networks; Facebook; privacy-preserving; photo sharing; JPEG

1. INTRODUCTION

With the wide-adoption of mobile devices equipped with high-resolution on-board cameras, online photo sharing has become an extremely easy and popular activity. Nowadays, users can conveniently share photos through OSNs such as Facebook and Google+, or dedicated photo sharing platforms such as Instagram. As these images contain significant amount of personal information, they are privacy-sensitive in nature. This has triggered the research on the problem of privacy protection arisen from online photo sharing.

Some works focused on designing access control protocols such that the shared photos can only be accessed by a selected group of users [2, 1, 3]. However, as pointed out by [7], the existing privacy controls are far from adequate, leading to severe information leakage when users fail to understand the complex privacy settings, or when the OSN cannot implement the access control polices correctly.

Another category of privacy protection schemes aims at encrypting the image data, before uploading them to the OSN for sharing. Essentially, stronger protection could be offered, as in this case even the OSN itself cannot get the information of the original images. A challenging issue under this framework, however, is the design of an end-to-end image encryption/decryption mechanism that is robust against the lossy operations conducted by the OSN. For instance, Facebook converts all uploaded images to JPEG, choosing quality factor (QF) adaptively without user input.

Along this line, Ra *et al.* proposed the P3 [5], a privacy-preserving photo sharing system, which splits an image into a public part and a private part. The public part is shared over the OSN in the plaintext, and the secret part is encrypted and stored in a cloud storage server. The introduction of an additional cloud server dramatically complicates the file management system. More importantly, the public part of P3 leaks significant visual information of the original image. Alternatively, Tierney *et al.* designed a system called Cryptagram, enabling users to encrypt photos with traditional block ciphers and then embed the encrypted bit stream into a JPEG file. However, the desirable properties of Cryptagram are realized at the cost of a significantly increased storage burden in the OSN, as the use of cover image results in significant file expansion. Furthermore, selective image encryption was employed to provide privacy protection for online photo sharing [9, 10]. Unfortunately, they are not robust against the lossy operations in Facebook.

In this work, we focus on the privacy protection issues of the online photo sharing over Facebook. For simplicity and without loosing generality, we here only consider the grayscale images. The proposed technique could be readily extended to the other OSNs and color images as well. We first conduct an in-depth study on the manipulations that Facebook performs to the uploaded images. With the awareness of such information, we suggest a DCT-domain image encryption scheme that is robust against these lossy operations. As validated by our experimental results, superior performance in terms of security, quality of the reconstructed images, and storage cost can be achieved.

MM '16 October 15–19, 2016, Amsterdam, The Netherlands.

DOI: http://dx.doi.org/10.1145/2964284.2967288

2. PROBE INTO FACEBOOK

It has been observed that almost all the existing OSNs apply various lossy operations to the uploaded images [5, 4]. To design a robust privacy-preserving photo sharing scheme, it is very important to know how the OSN manipulates the uploaded images. In this work, we focus on a representative OSN: Facebook, because 1) Facebook is one of the most popular OSNs that integrates image sharing. In the first quarter 2015 earnings announcement, Facebook revealed that it now has more than 1.44 billion monthly active users, and this number is still growing. 2) According to [4], the operations conducted by Facebook are much more complex than the other OSNs such as Google+ and Twitter. The results obtained over Facebook could be easily extended to other platforms. Though [4] investigated this problem, some of their results are not accurate and many are no longer valid, probably due to the system update of Facebook.

Facebook converts the uploaded images into pixel domain, regardless of their original formats, and then applies the following types of operations when necessary: 1) resizing, and 2) JPEG compression, and 3) enhancement.

Resizing It is found that Facebook performs resizing when the resolution of the upload image is too large. As resizing could cause significant information losses, it is crucial to limit the resolution of the uploaded image such that no resizing is performed. To know the maximum tolerable resolution, we upload a series of images of different resolutions to Facebook, and compare with the downloaded versions. We observe that when both the length and width dimensions are less than 2048 pixels, the image resolution will remain intact. Otherwise, resizing is conducted.

JPEG Compression All the uploaded images are applied to an additional round of JPEG compression, in which the users *cannot* choose the value of QF. As the distortion caused by JPEG compression is directly related to the employed QF, it is of great importance to know the mechanism of assigning QF for different images. Some prior works [4] only roughly studied this issue, and claimed that the employed QF values are in the range of [76, 86].

To gain more accurate and up-to-date information, we upload all the 1338 images from UCID-v2 [6] to Facebook, and then extract the quantization tables from the downloaded images. It is observed that the quantization tables match exactly with the ones included in the International JPEG Group standard [8]. The employed QF values are image dependent and range from 71 to 92, as can be seen from Fig. 1. Generally, for images with rich amount of activities, the QF values tend to be low, while for those with large portion of homogenous regions, the QF values seem to be high. This is reasonable to better limit the sizes of the compressed files. We also have another very important observation: for encrypted images, no matter which type of encryption algorithm is applied, the QFs adopted are *consistently* 71. Essentially, Facebook treats encrypted images as high-activity ones, and apply the lowest QF.

Enhancement Filtering Besides JPEG compression, Facebook also applies additional enhancement filtering to improve the appearance of the images. Unfortunately, we find that it is very challenging to know such enhancement filtering exactly, as they are applied locally and are highly adaptive. The designed privacy-preserving photo sharing scheme should be robust to these unknown lossy operations as well.

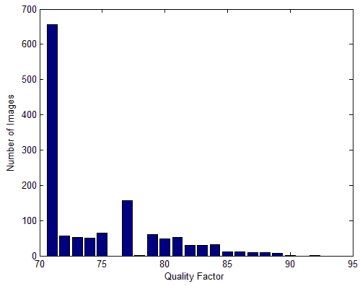


Figure 1: Distribution of the employed QFs.

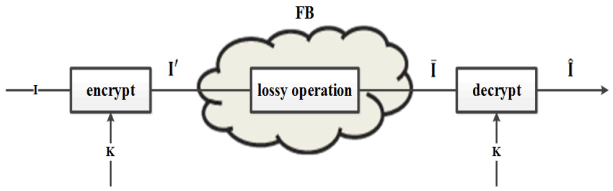


Figure 2: Schematic diagram of the proposed privacy-preserving image sharing scheme.

3. SYSTEM MODEL AND DESIGN GOALS

The schematic diagram of the whole system is depicted in Fig. 2. A user Alice first encrypts the original image $\mathbf{I}$ into $\mathbf{I}'$ by using an encryption function `Encry`

$$\mathbf{I}' = \texttt{Encry}(\mathbf{I}, K) \tag{1}$$

where K is the secret key shared between Alice the recipient Bob. Alice then uploads the encrypted $\mathbf{I}'$ to Facebook for sharing purpose. Facebook converts the received $\mathbf{I}'$ to the pixel domain, and then applies JPEG compression. As discussed in Section 2, the employed QF for JPEG compression is 71. The processed image $\bar{\mathbf{I}}$ of JPEG format is stored in the Facebook server, and shared over the Internet. At the client side, Bob downloads $\bar{\mathbf{I}}$ from Facebook and applies the following decryption function to produce an estimated $\hat{\mathbf{I}}$.

$$\hat{\mathbf{I}} = \texttt{Decry}(\bar{\mathbf{I}}, K) \tag{2}$$

The key components of the privacy-preserving photo sharing scheme are the encryption and decryption functions `Encry` and `Decry`. The design goals are summarized as follows.

i) *Reasonably high level of security*: Those parties that do not access the secret key K cannot get meaningful information about the original image. This is the fundamental requirement for any privacy-preserving photo sharing scheme.

ii) *High quality of the reconstructed image*: The reconstructed image $\hat{\mathbf{I}}$ should be of high quality.

iii) *High efficiency of storage on Facebook*: We also would like to minimize the storage cost in Facebook, as the number of photos shared becomes astonishingly large. This is also related to the bandwidth consumption in the client's side.

4. PROPOSED IMAGE ENCRYPTION AND DECRYPTION

After knowing the processing conducted over Facebook, we now present the key components of the proposed privacy-

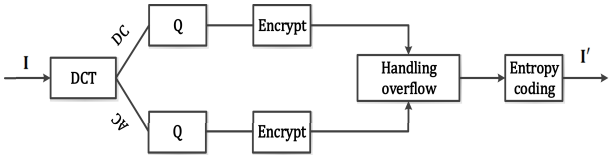


Figure 3: Processing-aware Image Encryption

preserving photo sharing scheme: image encryption and decryption modules. We should bear in mind that these two modules should be designed in a way to be robust against the lossy operations performed in Facebook. As the dominating degradation source in Facebook is the JPEG compression carried out over the DCT domain, it is natural to choose the DCT domain for performing encryption, so as to better limit the distortion incurred. Note that the encrypted file will be converted into pixel domain before JPEG is applied by Facebook. Such DCT-pixel domain conversion could cause severe pixel overflow problem that has to be tackled to ensure the high-quality of the reconstructed images.

Fig. 3 depicts the design of our proposed image encryption scheme. We first divide image $\mathbf{I}$ into a series of non-overlapping blocks of size 8×8. Let $\mathbf{B} = \{B_{r,s}\}$ be a generic 8×8 pixel block, where $0 \leq r, s \leq 7$. For each $\mathbf{B}$, we apply the DCT transform and obtain the DCT coefficient block $\mathbf{D} = \{D_{i,j}\}$, where $0 \leq i, j \leq 7$. Clearly, $D_{0,0}$ corresponds to the DC component, while the remaining ones are AC components. We then quantize $D_{i,j}$ with the JPEG quantization table associated with QF=71. As will be proved in our forthcoming work, this strategy of choosing the quantization table minimizes the end-to-end distortion $d = \|\mathbf{I} - \hat{\mathbf{I}}\|_2^2$. Due to the different characteristics of DC and AC components, we encrypt them separately, as detailed below.

4.1 Encryption of DC coefficients

We can easily get the quantized DC coefficients of all the blocks of the input image $\mathbf{I}$, and form a DC vector

$$\mathbf{DC} = (DC_0, DC_1, \cdots, DC_{n-1})' \tag{3}$$

where n is the number of blocks in $\mathbf{I}$. We reshape the vector $\mathbf{DC}$ into a 2-D block having four columns and $\lceil n/4 \rceil$ rows. We then perform two key-driven cyclical shift operations to the resulting 2-D block of DC components, and read out the data in raster-scan order to obtain the permuted DC vector $\tilde{\mathbf{DC}} = (\tilde{DC}_0, \tilde{DC}_1, \cdots, \tilde{DC}_{n-1})$. The key vectors controlling the column and the row shift offsets for $\mathbf{DC}$ are obtained from a stream cipher, which implies that the employed key vectors could be different, even for the same image encrypted at different sessions. Note that such permutation operations can be realized via circular shifts, which can be easily implemented in either hardware or software.

4.2 Encryption of AC coefficients

In JPEG compression, AC coefficients are encoded by run-length coding, implying that the coding efficiency is determined by the locations of none-zero coefficients and their code lengthes. To minimize the storage overhead of the encrypted images, our strategy of encrypting the AC coefficients is to randomize the non-zero coefficients using a stream cipher, while maintaining the location and the binary length of each non-zero AC coefficient. The steps of performing the AC coefficients encryption are as follows.

Step 1: Generate a binary key stream $\mathbf{KA}$ using the secret key K through a stream cipher, and partition it into $\mathbf{KA} = \mathbf{KA}(0), \mathbf{KA}(1), \cdots$, where each $\mathbf{KA}(i)$ is of length 12 bits.

Step 2: Convert each non-zero AC coefficient AC_i into a bit sequence $\mathbf{W}_i$ of length l_i.

Here, the way of converting the bit sequence is the same as that done by JPEG.

Step 3: Encrypt AC_i by bit-wise XORing $\mathbf{W}_i$ with the first l_i bits of $\mathbf{KA}(i)$, i.e.,

$$\mathbf{W}_i' = \mathbf{W}_i \oplus \mathbf{KA}_{1 \rightarrow l_i}(i) \tag{4}$$

where $\mathbf{W}_i'$ is the encrypted version of $\mathbf{W}_i$ and $\mathbf{KA}_{1 \rightarrow l_i}(i)$ represents the first l_i bits of $\mathbf{KA}(i)$. The reason why we assign 12 bits for each $\mathbf{KA}(i)$ is to achieve better robustness. The lossy operations in Facebook could make some of the zero AC coefficients be non-zero, or may even change the lengthes of some non-zero AC coefficients. To address the challenging issue of achieving synchronization between the encoder and the decoder, we associate each AC coefficient, regardless zero or non-zero, with a key stream $\mathbf{KA}(i)$ of length 12, which is longer than the maximum length of any AC coefficient [8]. It can be easily seen that the local desynchronization only affects the decryption/decoding of the local AC coefficient; but such error will not propagate to influence the subsequent AC coefficients.

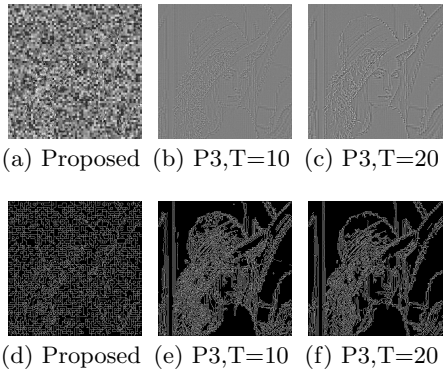
Step 4: Convert each binary sequence $\mathbf{W}_i'$ into signed decimal number, denoted as AC_i'.

A desirable property of the proposed AC encryption scheme is that the positions and the lengths of the non-zero AC coefficients are kept intact. This property makes the encrypted file readily compressible by Facebook without the need of accessing the secret key K.

4.3 Handling pixel overflow

As mentioned earlier, Facebook converts the uploaded images to the pixel domain, before applying JPEG compression. Due to the DCT-domain randomization, it is very likely that some of the pixel values are outside the range [0,255] after converting, causing the pixel overflow problem. This could severely affect the quality of the reconstructed images in the user end. To solve this challenging issue, we here propose a solution through DCT coefficient shrinkage.

Specifically, after applying the above DCT-domain encryption, we mimic the operations of Facebook by converting the encrypted blocks to the pixel domain. We then record the blocks that contain the overflow pixels in a location map $\mathbf{M} = \{M_{i,j}\}$. In general, the location map is sparse and hence compressible. Let $\mathbf{D} = \{D_{i,j}\}$ be a generic DCT block having the overflow problem. Our strategy of solving the overflow is to multiply each $D_{i,j}$ with a shrinkage factor $\alpha_{i,j} \in [0, 1]$, reducing the energy of the whole block. Smaller $\alpha_{i,j}$ can more effectively eliminate the overflow problem; but at the same time kills more small AC elements to zeros, leading to another type of distortion. The determination of $\alpha_{i,j}$ can be formulated as an optimization problem to balance the distortion caused by the overflow and the one caused by killing small AC elements. We leave the detailed discussion on the optimal determination of $\alpha_{i,j}$ to our future work. In this work, we set $\alpha_{0,0}$=0.20, and $\alpha_{i,j}$=0.40. The processed



(a) Proposed (b) P3,T=10 (c) P3,T=20

(d) Proposed (e) P3,T=10 (f) P3,T=20

Figure 4: Visual security comparison with P3 for the test image Lena. The first row gives the images that are exposed to Facebook. The second row represents the versions obtained by edge detection.

DCT blocks are subject to entropy coding to produce the final encrypted and compressed file $\mathbf{I}'$

4.4 Decryption function Decry

After downloading the encrypted JPEG file from Facebook, Bob aims to decrypt and decode it to the pixel domain with the assistance of the secret key K. The encrypted file can be parsed into blocks of DCT coefficients. We first check the location map $\mathbf{M}$, in which we record the overflow blocks. If the current block does not contain any overflow pixels, we can decrypt the DC and AC components by applying the reverse operations as those performed in the encryption stage. While if the current block is recorded in $\mathbf{M}$, we first need to divide the shrinkage factors applied to DC and AC coefficients, and then apply the decryption as a normal block. After decrypting all the DC and AC components, the pixel values can be recovered by applying the inverse DCT.

5. EXPERIMENTAL RESULTS

In this section, we experimentally evaluate the performance of our proposed privacy-preserving photo sharing scheme. The test set is composed of 100 images randomly selected from UCID-v2 [6].

We first compare the visual security of our proposed scheme with P3, while leaving the theoretical security analysis for our future work. In Fig. 4, we show the images that are accessible to Facebook for the test image Lena. As can be seen, Facebook can hardly get any meaningful information from the encrypted version of our proposed scheme, even after edge detection. In contrast, for P3, the public part stored in Facebook reveals significant visual information of the original image. The edge detected version could be treated as a sketch of great fidelity, and the quality is improved by increasing the parameter T. Similar observations can be obtained for the other test images as well.

We then investigate the quality of the reconstructed images. In Table 1, we show the PSNR results of offline JPEG with QF=71, our proposed method, P3, and the one based on secure JPEG [10]. Compared with the offline JPEG, the PSNR drop of our method is around 1.5 dB on average, due to the lossy operations performed by Facebook. In contrast, the method in [10] leads to severe PSNR degradation, with average PSNR drop being 7.94 dB. P3, on the other hand, has its inherent advantages in terms of the quality of the reconstituted images because most of the parts are stored in an error-free cloud server. The PSNR gap between our method and P3 is not big; less than 1 dB on average. It should be noted that the higher reconstruction quality of P3 is achieved through the introduction of a third-party cloud server, which may not be realistic in many situations.

Table 1: The Comparison of Reconstruction Quality in terms of PSNR. Here, Average is obtained by averaging the results of the 100 images from UCID.

	JPEG	Proposed	[10]	P3
Lena	37.40	36.32	28.85	36.84
Baboon	30.64	30.63	24.51	30.63
Goldhill	35.27	34.67	29.24	35.24
Truck	32.96	32.75	28.31	32.95
Elaine	33.94	33.61	28.49	33.81
Airplane	38.58	37.18	29.10	38.23
Couple	35.34	34.15	27.20	34.96
Average	34.56	33.02	26.62	34.12

Table 2: The Comparison of Storage Overhead

	Proposed	P3(T=10)	[10]
Lena	3.2%	20.2%	5.7%
Baboon	1.4%	13.8%	2.2%
Goldhill	3.7%	15.2%	8.8%
Truck	2.3%	12.6%	3.3%
Elaine	3.1%	15.9%	5.2%
Airplane	6.0%	25.5%	17.5%
Couple	1.9%	18.0%	3.3%
Average	2.0%	16.0%	5.6%

We finally study the storage overhead in the Facebook, which becomes an important evaluation criterion due to the huge number of shared images. In Table 2, we show the overhead incurred by the privacy protection for different methods, where the overhead is calculated with respect to the offline JPEG with QF=71. On average, the overhead of our proposed method is 2%, which is only 1/8 and 1/2.8 of the counterparts given by the competing P3 and secure JPEG.

6. CONCLUSION

In this work, we have addressed the problem of designing privacy-preserving, high-fidelity, and storage-efficient photo sharing scheme over Facebook. Based on a DCT-domain image encryption method robust to various lossy operations conducted by Facebook, we have achieved superior performance in terms of security, quality of the reconstructed images, and storage cost, compared with the state-of-the-art competing solutions.

7. ACKNOWLEDGMENTS

This work was supported in part by the Macau Science and Technology Development Fund under grants FDCT/009/2013/A1, FDCT/046/2014/A1, in part by the Research Committee at University of Macau under grants MYRG2014-00031-FST, MYRG2015-00056-FST,and in part by the National Science Foundation of China under grant 61402547.

8. REFERENCES

[1] L. A. Cutillo, R. Molva, and M. Önen. Privacy preserving picture sharing: Enforcing usage control in distributed on-line social networks. In *Proc. SNS'12*, page 6. ACM, 2012.

[2] 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 *Proc. CHI'12*, pages 377–386. ACM, 2012.

[3] 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 *Proc. FAST'14*, pages 89–103. USENIX Association, 2014.

[4] J. Ning, I. Singh, H. V. Madhyastha, S. V. Krishnamurthy, G. Cao, and P. Mohapatra. Secret message sharing using online social media. In *Proc. CNS'14*, pages 319–327. IEEE, 2014.

[5] M.-R. Ra, R. Govindan, and A. Ortega. P3: Toward privacy-preserving photo sharing. In *Proc. NSDI'13*, pages 515–528. USENIX Association, 2013.

[6] G. Schaefer and M. Stich. UCID - An uncompressed colour image database. In *Proc. SPIE: Storage and Retrieval Methods and Applications for Multimedia*, volume 5307, pages 472–480, 2004.

[7] M. Tierney, I. Spiro, C. Bregler, and L. Subramanian. Cryptagram: Photo privacy for online social media. In *Proc. COSN'13*, pages 75–88. ACM, 2013.

[8] G. K. Wallace. The jpeg still picture compression standard. *IEEE Transactions on Consumer Electronics*, 38(1):xviii–xxxiv, 1992.

[9] L. Yuan, P. Korshunov, and T. Ebrahimi. Privacy-preserving photo sharing based on a secure jpeg. In *Proc. INFOCOM WKSHPS'15*, pages 185–190. IEEE, 2015.

[10] L. Yuan, P. Korshunov, and T. Ebrahimi. Secure jpeg scrambling enabling privacy in photo sharing. In *Proc. FG'15*, volume 4, pages 1–6. IEEE, 2015.