Light-Weight Selective Image Encryption for Privacy Preservation
Yi-Hui Chen, Eric Jui-Lin Lu, and Chu-Fan Wang


View online: https://doi.org/10.11989/JEST.1674-862X.80327011

Articles you may be interested in

Follow JEST WeChat public account for more information
Abstract—To protect personal privacy and confidential preservation, access control is used to authorize legal users for safe browsing the authorized contents on photos. The access control generates an authorization rule according to each permission assignment. However, the general access control is inappropriate to apply in some social services (e.g., photos posted on Flickr and Instagram, personal image management in mobile phone) because of the increasing popularity of digital images being stored and managed. With low maintenance loads, this paper integrates the data hiding technique to propose an access control mechanism for privacy preservation. The proposed scheme changes the partial regions of a given image as random pads (called selective image encryption) and only allows the authorized people to remedy the random pads back to meaningful ones which are with similar visual qualities of original ones.

Index Terms—Fine-grained access control model, image access control, noise pads, privacy protection.

1. Introduction

In recent years, people used to obtain and store information from/up to the Internet since the widespread use of network technologies. The Internet is a public but insecure environment, which might make user’s private information vulnerable to malicious intruders during data transmission. To ensure that no illegal users can access the un-authorized contents, the access control model is one of the ways used in privacy preservation. Using the model, users can authorize people who have the rights to access their own digitized contents. After that, users can be granted or denied to resources after they log in the systems.

Nowadays, the massive use of multimedia arises several issues including privacy and confidentiality preserving[1][2]. Access control models for multimedia databases were proposed to provide safe browsing and publishing of multimedia[3][4]. In 2003, Bertino et al.[5] proposed a hierarchical access control model for video database systems. Based on hierarchical concepts, Thuraninsigham et al.[6][7] created a digest between the...
contents of the video and the corresponding description to facilitate the specification of permissions. Pickering et al.\cite{20} prevented the illegal copy by controlling the quality of videos. Later on, as for quality of services (QoS), Grosbois et al.\cite{8} and Phadikar et al.\cite{19} proposed schemes to modify the discrete wavelet transform (DWT) coefficients and discrete cosine transform (DCT) coefficients for providing different resolution and qualities of images to different users. In \cite{1}, Adam et al. proposed an access control model dedicated to the protection of digital libraries’ contents. In \cite{3}, Atluri and Chun proposed an access control model for geospatial data. Imaizumi et al.\cite{11} proposed a multi-layer encryption mechanism for Joint Photographic Experts Group 2000 (JPEG 2000) images used in privacy protection. However, all of them focused on accessing the whole digitalized contents, but not allowing just partial contents.

The control for encrypting the contents of multimedia is an important issue in recent years (e.g., masking out objects with violent scenes for TV show, hiding the sensitive information in a picture)\cite{4}. That is, one may prefer to access partial contents of an image rather than to the whole one for privacy protection, such as some leading services (e.g., Google Street View, EveryScape, and Mapjack), personal image management, and marketing promotion. Google Street View provides users to see the street view after they look up the location in the map; however, the service results in a high probability of leaking one’s private life even without meaning to do so\cite{2}. This has raised a need to de-identify individuals from the view of the street, but not encrypt the whole content of images. Also, it is useful for personal image management to just encrypt the sensitive regions of images as random pads. It is beneficial that the owner can see the non-sensitive region to recognize which pictures he looks for. Second, only the user who has rights can decrypt the encrypted regions to prevent the personal privacy leakage. As for marketing promotion on electronic publications, the partial interest words of bubbles or patterns in e-comics can be encrypted as random pads to attract readers to know what meaning contains in the pictures\cite{21}.

As for fine-grained access control for digitalized multimedia, Bouna et al.\cite{4} proposed an image access control model, which enables owners to authorize partial contents on images to users. The fine-grained access control model can specify the security rules based on the constraints, which provides a simple data representation model to properly describe the image description, and present the requirement of fine-grained control on images. It has a challenge that a significantly increasing number of pictures need maintenance, because lots of authorization needs maintenance. Pinto et al.\cite{21} changed the values of AC coefficients during the JPEG compression process to encrypt just the bubbles of e-comics, namely selective image encryption. After that, the encrypted areas can be restored to the original ones if the secret key is held. The partial image encryption keeps two advantages\cite{21}. First, the computational time can be reduced because only partial contents are required to encrypt. Second, the regions are selected by owners to encrypt partial contents according to their wishes. To achieve partial encryption, Pinto et al. proposed a selective encryption scheme\cite{21}, in which the encrypted regions are controlled by coefficients. However, it is not sophisticated; that is, some regions are not encrypted but the ones are exactly matched the correct ones to be encrypted as the wish. As for the fine-grained encryption scheme, Chen et al.\cite{7} proposed an almost lossless encrypted scheme to achieve fine-grained encryption on partial contents of images. However, it requires an extra procedure to embed the locations of the region of interest (ROI) into the image for later indicating the position of decryption. The sensitive region might leak out personal privacy information, called ROI. To meet the requirement, in this paper, we proposed a low cost model for selective image encryption without maintaining any authorization rule for personal image management. ROI is encrypted with a secret key, namely the encryption procedure. Later on, the users can decrypt ROI only if they own the secret key, called the decryption procedure.

This paper is organized as follows: Related works are briefly described in Section 1. In Section 2, the details of the proposed scheme are presented, including the encryption and decryption procedures. The experimental results are shown in Section 3. Finally, conclusions and directions for future work are drawn in Section 4.
2. Proposed Scheme

The proposed scheme is an improved version of the scheme in [8], and divides into three procedures, namely ROI processing, encryption procedure, and decryption procedure, which are described in sub-sections 2.1, 2.2, and 2.3, respectively.

2.1. ROI Regions Processing

In the proposed method, ROI and non-ROI must be recognized for later decryption. However, the traditional location map is with a big size for embedding into the image. Actually, the pixels in ROI are consecutively beneficial for data compression. Thus, we use the Peano mask tree (PM-tree) \cite{27} to compress the location map.

For example, scanning the whole image, the sensitive regions are tagged as ROI and non-ROI. Also, pixels in ROI and non-ROI are marked as 1 and 0, respectively, as shown in Fig. 1. The Peano count tree (P-tree) is generated as shown in Fig. 2 (a) according to Fig. 1.

First, the root of P-tree is the number of 1s in Fig. 1. After dividing Fig. 1 into four equal size pieces, the encoder accumulates 1s in each piece individually. If the number of 1s is zero or with the same as the size of the piece, the node of Fig. 2 (a) is treated as a leaf node; otherwise, it continues to divide into four pieces until it is a single pixel. After that, the P-tree is transformed into the PM-tree as shown in Fig. 2 (b), where the non-terminal nodes are replaced with \( m \). The terminal nodes are replaced with 1 and 0 if the values of the leaf nodes are larger than 1 and equal to 0, respectively.

Finally, the PM-tree is traced by using the pre-fix order to generate a bit string, where the value of node is replaced with 0, 10, and 11, if the value is 0 or 1, and \( m \), respectively.

2.2. Encryption Procedure

Note that the pixel is checked whether it is embeddable with (1) to avoid the overflow problem after encryption, where (1) is to get the most minimum pixel whose value after decrypting is larger than 255:

\[
\text{Arg min}(p_x + 1 + 3 \times (p_x \mod \alpha)) > 255
\]  

(1)

where \( p_x \) is the \( x \)th pixels of the image and \( \alpha \) is a digit larger than 3 as the encryption factors in a range of \([a, b]\), and \( \alpha \) is used as a private secret value to control the mosaic strength of ROI. Different pixels are encrypted with different encryption factors chosen by a secret key. If the pixel is not satisfied (1), it is judged as a to-be encrypted pixel, and the side information of it is recorded as 1; otherwise, 0.

The side information is combined and encoded as a bit sequential (bSQ) file. Later on, the bSQ file is compressed by the PM-tree \cite{27}. The pixel \( p_x \) is used to embed the bits of the compressed codes. The pixel is transformed into a bit stream. The data is embedded into the pixel \( p_x \) with (2):

\[
\begin{array}{cccccccc}
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
\end{array}
\]  

Fig. 1. Authorization region example for a given image with the size of 8×8.

\( \begin{array}{c}
27 \\
11 \ 0 \ 16 \ 0 \\
4 \ 2 \ 4 \ 1 \\
1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \\
\end{array} \)

(a) P-tree

\( \begin{array}{c}
1 \ 1 \ 1 \ 1 \\
1 \ 1 \ 1 \ 1 \\
1 \ 1 \ 1 \ 1 \\
1 \ 1 \ 1 \ 1 \\
\end{array} \)

(b) PM-tree

Fig. 2. Examples of P-tree and PM-tree after compressing data in Fig. 1: (a) P-tree and (b) PM-tree.
\[ p^* = p_x + \text{sign}(\lambda) \times (\lfloor |\lambda| \mod 2 \rfloor - |\lambda|) \]  

where \( s \) is the extracted compression codes from bits of bSQ, \( \lambda = s - (p_x \mod 3) \), \(|\lambda|\) is the absolute value of \( \lambda \), and \( \text{sign}(\cdot) \) is the function to get the value 1 or -1 if \( \lambda \) is a positive digit or negative one.

If the pixel is with the underflow problem after data embedding, the original pixel is plus one and do again. In the same way, if it is with the overflow problem, the pixel is minus one before data embedding. Next, the ROI pixels are encrypted with (3):

\[ p^*_x = p^*_x + 3 \times (p^*_x \mod \alpha). \]  

### 2.3. Decryption Procedure

During decoding, the hidden \( s \) is extracted by (4):

\[ s = p_x^* \mod 3 \]  

Thus, the locations of ROI are extracted. Next, the extracted codes are the compression codes used to reconstruct the PM tree. The PM tree can be transformed into P-tree and then be the bQS file. Finally, ROI can be recognized. According to the extracted ROI, the encrypted regions can be decrypted with (5):

\[ p^*_x = p^*_x - 3(p^*_x \mod \alpha) \quad \text{and} \quad p^*_x - 3(\alpha - 1) \leq p^*_x \leq p^*_x. \]

The \( p^*_x \) value can be decrypted as \( p^*_x \) satisfying two condition shown in (5). To meet the above two conditions, only one pixel is retrieved as decrypted results.

For clarity, an example is provided to demonstrate the encryption and decryption procedures. If the pixel \( p_x \) is 53 and the secret data is 1, the stego-pixel is 52 derived from \( \lambda \) put into (2) \( \lambda = 53 \mod 3 = 1 \), \( \text{sign}(\lambda) = -1 \), and \( p^*_x = 53 + (-1) \times 1 - 0 = 52 \). If the value of \( \alpha \) is 7, the encrypted pixel \( p^*_x \) is 61 \( (p^*_x = 52 + 3 \times (52 \mod 7) = 61) \). The receiver can extract the hidden data \( s \) with (4). In the case, the extracted hidden data \( s \) is 1, since \( (61 \mod 3) = 1 \). During the decryption procedure, to find the pixel \( p^*_x \), it must hold the condition \( p^*_x = p^*_x - 3(p^*_x \mod \alpha) \) and fall into the range of \([p^*_x - 3(\alpha - 1), p^*_x]\). In this case, the pixel value is obtained as 52 satisfying the condition and in the range of \([43, 61]\). The given \( p_x \) is 111 and secret data \( s \) is 2 to output the encrypted pixel, extracted hidden secret, and decrypted pixel, which are listed in Table 1. The pixel is judged as un-embeddable if the pixel satisfies (1).

### 3. Experimental Results

In the experiments, the test images, street, people, and street2 are shown in Figs. 3 (a) to (c) to evaluate the performances. After data embedding, the corresponding embedded images are shown in Figs. 3 (d) to (f), respectively. The visual quality on average is higher than 65 dB; thus, it is difficult to realize the differences between the embedded images and the original ones. The images are encrypted according to different sizes of ROI as shown in Figs. 3 (g) to (i) with the encrypted factor \( \alpha \), which is a prime digit for an encryption pixel chosen by the secret key in the range of \([3, 47]\). After encryption, it is very difficult to recognize the original content with naked eyes. After decrypting the encrypted image, the decrypted results are shown in Figs. 3 (j) to (l) which are the same as the embedded images.

The size of ROI of Figs. 3 (g) to (i) and the length of compression codes are listed in Table 2. Also, the visual qualities of Figs. 3 (j) to (l) evaluated by the peak signal-to-noise ratio (PSNR) are listed in Table 2, too. The results show that the larger size of ROI results in lower visual quality. In addition, if ROI region is dispersed, the length of

---

**Table 1: Example of encryption and decryption procedures**

<table>
<thead>
<tr>
<th>( p_x )</th>
<th>( s )</th>
<th>( \rho^*_x )</th>
<th>( \rho^*_x )</th>
<th>Extracted ( s )</th>
<th>Decrypted ( \rho^*_x )</th>
</tr>
</thead>
<tbody>
<tr>
<td>53</td>
<td>1</td>
<td>52</td>
<td>61</td>
<td>1</td>
<td>52</td>
</tr>
<tr>
<td>111</td>
<td>2</td>
<td>110</td>
<td>125</td>
<td>2</td>
<td>110</td>
</tr>
</tbody>
</table>
After decrypting, the visual quality of the embedded image keeps intact, which is similar to the original one.

4. Conclusions

In this paper, a selective image encryption technique is proposed. The model is useful for preserving the personal privacy with low cost maintenance. The sensitive regions can be protected after data encryption. Only the people who get the key could decrypt the sensitive region. It could be applied to the image management in cloud storage.
References


Yi-Hui Chen received her Ph.D. degree in computer science and information engineering in 2009 from the National Chung Cheng University, Jiayi. From 2009 to 2010, she worked at IBM’s Taiwan Collaboratory Research Center, Taipei as a research scientist. Currently, she is with the Department of Information Management, Chang Gung University, Taichung as an associate professor and also with the Kawasaki Disease Center, Kaohsiung Chang Gung Memorial Hospital, Kaohsiung as an associate researcher. Her research interests include multimedia security, semantic web, text mining, and multimedia security.

Eric Jui-Lin Lu received the B.A. degree from the National Chiao-Tung University, Taichung in 1982. Later on, he received his MSBA degree from San Francisco State University, San Francisco in 1990. He received his Ph.D. degree in computer science from Missouri University of Science and Technology (formerly University of Missouri-Rolla), Missouri in 1996. He is currently a professor with the Department of Management Information Systems, National Chung Hsing University. His research interests include machine learning, image and semantic web, and multimedia security.

Chu-Fan Wang received the M.S. degree from National Chung Hsing University in 2013. Currently, he is working with Trend Micro, Taipei. His research interests include multimedia security and information security.