SCRAMBLING-BASED TOOL FOR SECURE PROTECTION OF JPEG IMAGES

Pavel Korshunov and Touradj Ebrahimi

Multimedia Signal Processing Group, EPFL, Lausanne, Switzerland

ABSTRACT

JPEG scrambling tool is a flexible web-based tool with an intuitive and simple to use GUI and interface to secure visual information in a region of interest (ROI) of JPEG images. The tool demonstrates an efficient integration and use of security tools in JPEG image format by an example of scrambling privacy filter, which enables a variety of security services such as confidentiality, integrity verification, source authentication, and conditional access.

Index Terms— Secure JPEG, demonstration, scrambling, web interface

1. INTRODUCTION

The success of digital imaging applications is in part due to the development of effective standards such as JPEG [1] and JPEG 2000 [2]. JPEG is one of the early standards and is *de facto* the most popular format for storing and compressing images thanks to its efficiency and low complexity. JPEG 2000 is a more recent standard for still image coding, offering efficient image compression, progressive transmission, seamless scalability, region of interest coding, and error resilience. However, even though JPEG 2000 outperforms JPEG in terms of compression, JPEG is still the most popular format in large variety of imaging applications.

Digital imaging applications and image sharing platforms are expanding into various aspects of the daily life, increasing public concerns regarding the security and visual privacy protection of image data. The ease with which digital images can be manipulated, copied, and distributed at negligible costs calls for an adequate content protection, authentication, data integrity, and privacy protection. Recognizing that security and privacy are the major issues in many imaging applications, JPEG committee created Secure JPEG 2000 or JPSEC specifications, which became an International Standard in 2006. The purpose of JPSEC is to provide a framework for secure imaging in JPEG 2000. In this paper, taking into account the fact that JPEG is still very important and will continue to be widely used in the foreseeable future, we propose to use a similar approach to JPSEC for secure protection of sensitive regions in JPEG images. The goal is to make it possible for JPEG to have the same security services that JPSEC enables for JPEG 2000. To demonstrate such possibility a prototype of JPEG Scrambling tool was implemented and presented at JPEG meeting in London, 2013.

In this paper, a JPEG Scrambling tool is presented, as a web-based service, with a simple and intuitive GUI and publicly accessible interface[1]. Scrambling technique [3][4] was implemented in JPEG Scrambling tool as an example of security and privacy protection services that can be incorporated in JPEG. Scrambling is an attractive alternative for protection of image and video content while keeping complexity low. Scrambling can be effectively applied on the quantized DCT coefficients of a selected region of an image, as proposed by Dufaux and Ebrahimi in [3]. At the decoder side, authorized users can perform the reversed operation, unscrambling the coefficients, thus, allowing a fully reversible process for authorized users. Unauthorized users without the knowledge of a secret key used for scrambling would not be able to recover the protected image region, but will be able to still decode the image using a legacy JPEG decoder, with the selected region of interest scrambled.

JPEG Scrambling tool demonstrates the following concepts that are important in various practical applications, including video surveillance, media sharing social networks, and medical imaging:

- Conditional access to sensitive regions in JPEG compressed images.
- JPEG compressed image integrity verification.
- Adaptation of scrambling technique with a simple and intuitive GUI, and web-based usage scenarios.

2. IMPLEMENTATION CHOICES

In principle, there are three ways a JPEG image can be scrambled:

1. by scrambling the values of image pixels before JPEG compression;
2. by scrambling the values of quantized coefficients before entropy coding;
3. by scrambling the bits of entropy coded coefficients.

This work has been conducted in the framework of EC funded Network of Excellence VideoSense. Special thanks for the help in developing web interface of JPEG Scrambling tool to Taha Elgraini.

[1]The web-demo can be found here: http://ltslinux18.epfl.ch/scramble

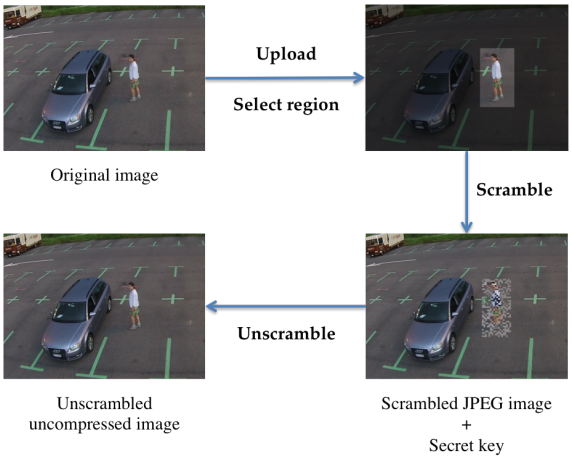


Fig. 1: The process of scrambling and unscrambling of an image using JPEG Scrambling tool.

Option 1 is not efficient as it results in an image that is structurally similar to noise, which renders compression inefficient. Also, it does not work on JPEG compression, since it is lossy and there will always be residual errors after unscrambling. Option 3 requires an additional step to make sure the generated bitstream is decodable by a conventional decoder, and it is too complex with little added value. Therefore, Option 2 is the best choice for implementation of scrambling algorithm in JPEG. The algorithm can be implemented in two ways: (i) either by encrypting the bits or (ii) by generating a pseudo random suite of 1s and 0s and flipping the value of the bits that correspond to 1 in the suite as originally proposed in [3]. In the second case, the seed for pseudo random number generator is encrypted to make sure that the process cannot be reverse engineered by somebody who knows the algorithm.

In JPEG Scrambling tool, scrambling Option 2 is used in combination with the implementation choice (ii). The coefficients of DCT after quantization are represented in their binary mode with their signs flipped according to random value. A pseudo random generator uses a seed, which is encrypted and inserted as an APP marker in the header of the JPEG image. APP marker is a generic concept that is commonly used by JPEG encoders and decoders for inserting and retrieving additional information to and from JPEG image.

In summary, the encoder (the decoder follows the reversed steps) of JPEG scrambling tool follows these steps (see Figure 1 for illustration):

1. Select region of interest or the whole image and save the location information about the ROI. The information is inserted in the JPEG bitstream via APP marker.
2. During compression the processed image is scrambled by modifying the values of quantized DCT coefficients before entropy coding.
3. Scrambling is applied after quantization horizontally block by block according to the selected ROI.
4. A pseudo random binary sequence is generated and the signs of DCT coefficients are flipped for each value corresponding to 1 in the sequence. An optional scrambling level can determine the number of DCT coefficients that will be scrambled, with 64 being the highest level (all DCT coefficients are scrambled) and zero – the lowest (no DCT coefficients are scrambled).
5. The seed for pseudo random number generator is encrypted and together with ROI coordinates inserted in the JPEG bitstream via APP marker, which adds a few bytes of memory to the image size.
6. After scrambling, the image is entropy encoded following the standard JPEG process.

3. USAGE GUIDELINES

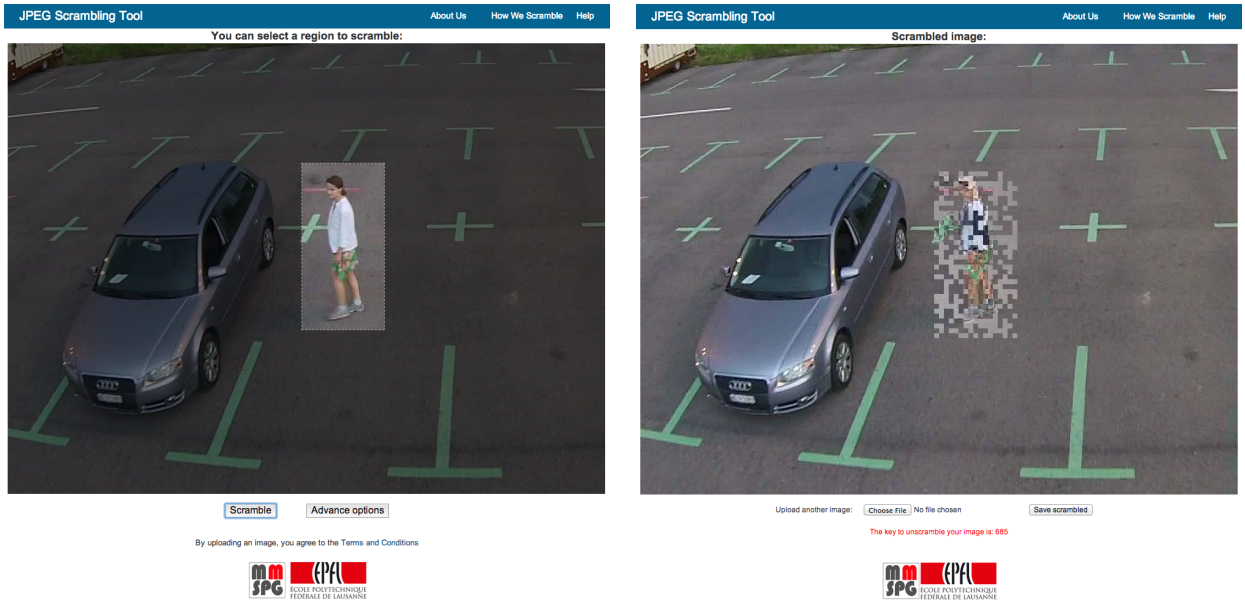
The JPEG Scrambling tool includes a scrambling-enabled implementation of JPEG encoder and decoder, which are based on IJG open source code of JPEG compression[2]. The tool runs as a web service. A simple and intuitive GUI was implemented to support different typical usage scenarios.

3.1. GUI design

The GUI implementation of JPEG Scrambling encoder provides a typical set of features for loading and displaying images, basic manipulation tools, and JPEG compression with scrambling option. Since the main part of scrambling algorithm is embedded in the JPEG encoding process, the GUI can be made simple and easy to use.

Figure 2 demonstrates screenshots of JPEG Scrambling tool GUI. Once the image is uploaded, it is shown on the screen, allowing to select a region of interest for scrambling with a mouse of a pointing device. If no ROI is selected, the

[2]http://www.ijg.org/



(a) Select ROI

(b) Scrambled image with a key

Fig. 2: Example screenshots of JPEG Scrambling tool web interface.

whole image will be scrambled. The pressing of 'Scramble' button will lead to the image with scrambled ROI to appear as shown in Fig. 2b. The secret key is displayed under the image. The scrambled image can be downloaded via 'Save scrambled' button. The interface therefore allows to export the scrambled image to other environments. For sake of simplicity, no further options such as the strength of the encoder or manual entry of secret key are shown in this simple GUI but can be available as advance options. The GUI is compatible with major platforms, devices and browsers such as smart phones, tablets and laptops.

The decoder interface is similar but in reverse order when compared to the encoder. To successfully unscramble an image, the decoder requires the corresponding secret key, otherwise the image would remain scrambled with unrecognizable visual information.

Web-based interface of JPEG scrambling tool (see Fig. 2) demonstrates how scrambling can be used to protect privacy in a typical online scenario, for example, in an image sharing service. The web implementation is kept simple and straightforward to show how easy it is for a user to upload a photo, select region that needs to be scrambled (protected), and obtain the resulted scrambled image with a secret key. When a scrambled image is uploaded, the interface requests for the secret key, and if it is correct, the image will be unscrambled displaying the original content. If key is not correct, the image will not be unscrambled. In the privacy protection context, it means that the identity of the person whose privacy is protected will not be visible when the wrong key is used, since access rights are not granted.

4. CONCLUSION

This paper presents a prototype of a JPEG scrambling tool that allows a user to securely protect selected regions of interest (e.g., for privacy reasons) by scrambling in the JPEG compression domain and assigning a key to the protected image. Another user in possession of such key will be able to recover the protected image by using the provided decoder. The prototype includes a simple and intuitive GUI and a web-based interfaces, demonstrating typical privacy protection usage scenarios.

5. REFERENCES

[1] G. Wallace, "The JPEG still picture compression standard," *IEEE Trans. on Consumer Electronics*, 1991.

[2] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The JPEG 2000 still image compression standard," *IEEE Signal Processing Magazine*, vol. 18, no. 5, pp. 36–58, Sep 2001.

[3] F. Dufaux and T. Ebrahimi, "Video surveillance using JPEG 2000," in *proc. SPIE Applications of Digital Image Processing XXVII*, Denver, CO, Aug 2004, vol. 5588, pp. 268–275.

[4] 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.