SECURE JPEG 2000 - JPSEC

Touradj Ebrahimi and Raphael Grosbois

Swiss Federal Institute of Technology – EPFL
CH-1015 Lausanne
Switzerland

ABSTRACT

The Joint Photographic Experts Group (JPEG) has recently rolled out a new still image coding standard called JPEG 2000. This standard integrates an efficient image compression scheme with functionalities required by multimedia applications, such as progressiveness up to lossless coding, region of interest coding, and error resiliency. Security is a concern in many applications, and therefore also a desired functionality. This paper provides readers with insights and examples of how to combine security solutions with JPEG 2000 compression. Tools for JPEG 2000 compressed image integrity, access control, and copyright protection are presented. They can be either applied to a JPEG 2000 codestream or directly integrated into the coding/decoding operations, resulting in a fully compliant JPEG 2000 image.

1. INTRODUCTION

With the expansion of digital imaging applications, it appears relevant to develop security tools to check the integrity of an image, to control its access by end-users, and to protect the intellectual property rights underlying its content.

The new still image compression standard JPEG 2000 has been defined for use in various multimedia applications, integrating features needed by these applications such as good quality at high compression ratios, seamless scalability, and region of interest coding, to mention a few [1].

Security being an important concern, JPEG 2000 is currently considering how to extend its specifications in order to integrate tools needed to secure images.

This paper reports the road map for such an effort, and presents examples of tools for securing JPEG 2000 images. The next section provides an overview of JPEG 2000 standard and its various parts. Section 3 presents a tool which can be used to check the integrity of JPEG 2000 compressed images. Section 4 describes an algorithm which can allow for conditional access to JPEG 2000 compressed images by either resolution or quality levels. Section 5 discusses how watermarking can be applied to JPEG 2000 compressed images. Finally, Sec. 6 concludes the paper.

2. JPEG 2000 OVERVIEW

JPEG 2000 specifications have been clustered in several parts, namely:

Part 1: contains the core system compression scheme and an optional light file format (jp2).

Part 2: provides additional tools for specific applications such as remote sensing, and a rich file format (jpx).

Part 3: contains a file format for compression of image sequences together with other associated data (mj2).

Part 4: provides conformance guidelines.

Part 5: contains reference software implementations in Java and C languages.

Part 6: contains a file format for efficient coding of compound images (jpm).

Part 7 of the standard is void, as its original objective as a technical report for hardware implementation guidelines was abandoned.

Other new parts have been initiated by JPEG committee and are under development:

Part 8: Secure JPEG 2000 (jpsec)

Part 9: Interactivity and Protocols (jpip)

Part 10: Volumetric data (jp3d)

Part 11: Wireless (jpwl)

This paper aims at providing a roadmap and discussion for potential solutions to Secure JPEG 2000 (JPSEC).

In a typical implementation of JPEG 2000 part I compliant encoder, discrete wavelet transform (DWT) is first applied to the source image. The transform coefficients are then quantized and entropy coded on a code-block by code-block basis, before forming the output codestream. The resulting bitstream is built using a post-compression rate control and possesses a packet structure.

It is worth mentioning that unlike many alternative coding schemes, JPEG 2000 compression can be both lossy and lossless, always produces an intrinsically progressive

bitstream, and can compress a region of interest with higher quality than the rest.

3. IMAGE INTEGRITY

The goal of image integrity is to identify any attack on an image. Since a modification in the spatial domain translates into a modification of wavelet coefficients, it turns out that, in the JPEG 2000 framework, the integrity check procedure can be applied in the wavelet domain. Note that, due to the space-frequency localization of wavelet bases, it is still possible to determine the position of an attack from the wavelet domain.

Here, we describe an image integrity approach first proposed in [2]. The technique is based on authentication of code-blocks in the compressed bitstream by application of a hashing function and use of digital signatures. An originality of this method is in the approach used for the inclusion of the hash value inside the bitstream. The compliance with JPEG 2000 part I syntax is maintained by the fact that any byte in the bitstream, after a termination marker, won't be read by a part I compliant JPEG 2000 decoder.

For the hash function the Secure Hash Algorithm (SHA [3]) was chosen to generate digests for the code-blocks bitstreams. Note that any other hash function (MD2, MD4, MD5 ...) can be used for this stage. By applying the SHA algorithm on the bit streams, digital signatures with a fixed length of 160 bits per code-block are produced. This signature is then encrypted by a public-key encryption such as the Rivest, Shamir, and Adleman algorithm (RSA [4]). Again, any other encryption algorithm can be used for this stage.

By construction, the arithmetic decoder in JPEG 2000 stops reading bytes from the bitstream when it encounters a termination marker (i.e. two bytes with value greater than 0xFF8F). This interesting feature is used to add extra bytes in the bitstream without affecting the syntax compliance. Indeed, any added bytes will be skipped by JPEG 2000 compliant decoders. The only precaution to take is in the number of added bytes since this increases the overall bit-rate.

While compressing an image, this operation is performed between the entropy coding and rate-allocation stages in the JPEG 2000 compression algorithm. If the compressed image is already available, the insertion is done by parsing the bitstream without complete decoding and encoding. At the decoder side, first the code-block's bitstream is retrieved and the bytes before the termination marker are isolated from those after this marker (i.e. encrypted hash value or signature). A new hash value is the computed and compared with the decrypted one from the bitstream. An attack is detected on the code-block, and therefore its corresponding spatial location in the image, when the encrypted hash value is missing or when it is not equal to that generated from the preceding bytes in the bitstream.

4. ACCESS CONTROL

Access control to an image is among important functionalities in secure imaging. One often desires to give access to end-users to view a small size (resolution conditional access) or a lower quality (quality conditional access) version of an image, while access to larger sizes or higher qualities are subject to special authorizations.

In this section, we present a tool which allows for conditional access to a JPEG 2000 compressed image either by resolution or by quality, while maintaining compliance with part I of the standard.

4.1. Resolution access control

Here, we present a method that allows preview of low resolution(s) of an image, while preventing the correct display of its higher resolutions. The idea is to introduce a pseudo-random noise in the image, such that the decoded image appears distorted for a decoder that does not know how to remove this noise. To *scramble* the high resolutions of the image, a known noise is added to high frequency wavelet coefficients (i.e. the high resolution levels). In order to have a very simple algorithm, the signs of the coefficients in each code-block are inverted pseudo-randomly. Note that this method modifies only the most significant bit-plane of the coefficients and can be performed on-the-fly during the entropy coding. A pseudo-random generator based on a linear congruential formula [5] and characterized by its *seed* value was used in the method presented here. The sign change takes place as follows. For each coefficient, a new pseudo-random value is generated and compared with a density threshold. If the pseudo-random value is greater than the threshold, the sign is inverted; otherwise the sign is left unchanged. In order to improve the security of this scrambling technique, a different seed can be used for each code-block. To communicate the seeds values to authorized users, they are encrypted, and inserted at the end of the corresponding bitstreams in a way similar to the method presented in the previous section.

Figure 1 provides the results obtained by scrambling resolutions 1 to 3 (resolution 0 is left intact) of an image that is compressed with three wavelet decomposition levels. We observe the qualities of the decoded images when disposing of the seed or not. Naturally in both cases, the first resolution is identically reconstructed, because no noise has been introduced. However, for higher resolutions, the error increases between the two corresponding image resolutions and the distortion obtained without the seed becomes increasingly visible.

A typical application for this method would be the browsing of a database of JPEG 2000 images where a

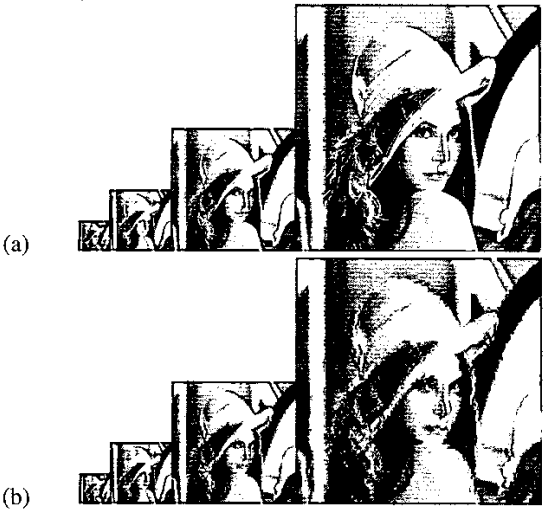

Figure 1: *Resolution progressive decoding of a codestream which has been scrambled starting from resolution 1. (a) Authorized user (b) Unauthorized user.*

certain number of high resolutions would have been scrambled. There, the client could select an image, download it, but won't be able to correctly display it before obtaining a key necessary to retrieve the random noise generator's seed.

4.2. Quality access control

This section provides for an access control on image qualities. Here, the idea is to introduce the pseudo-random noise directly in quality layers of the image. The advantage of this method is that it can be applied either directly on the bit stream or on the wavelet coefficients. With this method, an unauthorized user will still be able to display a rough quality of the image but such an image will contain too much distortion to be printed or used commercially.

Based on the same method as for image resolution scrambling, the bits belonging to high quality layers are pseudo-randomly inverted.

Figure 2 illustrates the application of the proposed method on a codestream containing three layers. We observe the visual quality of the decoded images when the scrambling method has been applied on the last layer, on the two last layers and on every layer. As expected, the visual quality decreases with the number of scrambled layers. One can imagine applications where the number of scrambled layers differ from one code-block to another, and consequently apply a finer control on the distortion introduced in the final image when decoding without knowing the seeds. This enables scrambling of regions of interest in an image.

Figure 2: *Quality layers scrambling of a codestream where three layers have been defined. Decoding by an unauthorized party (top left) original image. (top right) third layer is scrambled. (bottom left) second and third layers are scrambled (bottom right) all layers are scrambled.*

5. RIGHTS PROTECTION

Watermarking is often used as a solution to copyright protection. This section presents a watermarking technique applied to JPEG 2000 compressed domain for rights protection.

The method was first described in [6]. It is based on the addition of an *m-sequence-derived* pseudo-noise (PN), where m is the length of the image identifier to be inserted by the watermark. The identifier is considered as a bi-polar signature $\{b_k\}_{0\le k<m}$ where each symbol can take a value of 1 or -1, depending on whether the original identifier's bits are 0 or 1. Furthermore, two extra symbols e_1 and e_{-1}, respectively equal to 1 and -1, are appended to this identifier. They are used in the computation of a confidence threshold T_{conf} for watermark extraction.

The embedding formula for insertion of the symbol b_k onto the JPEG 2000 quantized wavelet coefficients is given by:

$$w_q'[i] = w_q[i] + \alpha[i].n_{b_k}[i]$$

Where $\alpha[i]$ is a positive strength factor and $n_{b_k}[i] = b_k.n[i]$ is the zero mean pseudo-noise generated from b_k. This noise signal is obtained as described below:

$$n[i] = \begin{cases} 1 - 2\lfloor r_2 + 0.5 \rfloor & \text{if } r_1 \le \delta \\ 0 & \text{if } r_1 > \delta \end{cases}$$

with r_1 and r_2 being two successive pseudo-random numbers and δ a density parameter.
The embedding strength factor is chosen as:

$$\alpha[i] = \left\lceil M_{dyn}[s].\rho_{act}[c].\rho_{mask}[i] \right\rceil$$

with M_{dyn} the maximum magnitude of the embedded noise for subband *s*, $\rho_{act} (0 \le \rho_{act} \le 1)$ the activity of the code-block *c*, and $\rho_{mask} (0 \le \rho_{mask} \le 1)$ a visual intra-band masking strength factor of coefficient *i*.
To avoid the quantization effects, the watermark is added after quantization (and before entropy coding).
By symmetry, the extraction is performed between the entropy decoding and dequantization modules, computing a correlation value for each identifier's symbol:

$$corr[b_k] = \sum_i \alpha'[i].n[i].w_q'[i] =$$
$$= \sum_i \alpha'[i].n[i].w_q[i] + \sum_i \alpha'[i].n^2[i].\alpha[i]b_k$$

where $\alpha'[i]$ is a positive detection factor. By taking the expectation of this equation and using the facts that w_q and n are uncorrelated and that $n^2[i]$ is either equal to 0 or 1, one finally obtains:

$$E(corr[b_k]) \simeq b_k \sum_i \alpha[i].\alpha'[i]$$

The estimated detection factor used for watermark extraction is:

$$\alpha'[i] = r_{M_{dyn}}[s].\rho'_{act}[c].\rho'_{mask}[i]$$

with $r_{M_{dyn}}$ the dual parameter of M_{dyn}, ρ'_{act} and ρ'_{mask} the activity of the decoded code-block and the visual masking, with values roughly similar to ρ_{act} and ρ_{mask} except when partial decoding of the image is performed.
Simulation results show that the above described watermarking scheme exhibits good performance in terms of robustness to transcoding and partial decoding.

6. CONCLUSIONS

This paper described several security solutions, namely, image integrity, conditional access, and watermarking, integrated within JPEG 2000 compression framework. It was shown that it is possible to combine security solutions with each others and with high compression efficiency, while maintaining full compliance with JPEG 2000 bitstream syntax. The ongoing JPSEC activity aims at formalizing such solutions in a newly defined part 8 of JPEG 2000 specifications.

ACKNOWLEDGEMENT

This work was partially supported by the European Project 2KAN. Please refer to http://www.2kan.org for more details about this project and its objectives.

REFERENCES

[1] A. Skodras, C. Christopoulos, T. Ebrahimi "The JPEG 2000 still image compression standard", IEEE Signal Processing Magazine , Volume: 18 Issue: 5 , Sept. 2001, Page(s): 36 -58.

[2] R. Grosbois, P. Gerbelot, T. Ebrahimi, "Authentication and access control in the JPEG 2000 compressed domain", In Proc. of the SPIE 46th Annual Meeting, Applications of Digital Image Processing XXIV, San Diego, July 29th -August 3rd, 2001.

[3] FIPS Publication 180, "Secure Hash Standard (SHS)", NIST, May 11, 1993.

[4] R.L. Rivest, A. Shamir, and L.M. Adleman, "A method for obtaining digital signatures and public-key cryptosystems", Communications of the ACM (2) 21, 1978, Page(s): 120-126.

[5] Donald Knuth, "The Art of Computer Programming", Volume 2, Section 3.2.1.

[6] R. Grosbois, T. Ebrahimi, "Watermarking in the JPEG 2000 domain", In Proc. of the IEEE Workshop on Multimedia Signal Processing (MMSP), October 3-5, 2001.