A QUANTITIVE SEMI-FRAGILE JPEG2000 IMAGE AUTHENTICATION SYSTEM

Qibin Sun[1], *Shih-Fu Chang*[1], *Maeno Kurato*[2] *and Masayuki Suto*[2]

[1] Department of E.E., Columbia University, New York City, NY10027, USA
[2] Oki Electric Industry Co., Ltd. Gunma 370-8585, Japan

ABSTRACT

In this article, we propose a novel integrated approach to quantitative semi-fragile authentication of JPEG2000 images under a generic framework which combines ECC and PKI infrastructures[1][2]. Firstly acceptable manipulations (e.g., re-encoding) which should pass authentication are defined based on considerations of some target applications. We propose a unique concept of *lowest authenticable bit rate* – images undergoing repetitive re-encoding are guarantee to pass authentication provided the re-encoding rates are above the lowest allowable bit rate. Our solutions include computation of content-based features derived from the EBCOT encoding procedure of JPEG2000, error correction coding of the derived features, PKI cryptographic signing, and finally robust embedding of the feature codes into image as watermarks.

Keywords digital signature, PKI, watermarking, image authentication, JPEG2000, error correction coding

1. INTRODUCTION

Semi-fragile image authentication deals with verifying authenticity of a received image while allowing some acceptable manipulations. In [3], a semi-fragile authentication watermarking solution was proposed for JPEG images. JPEG-specific invariant features are identified and used for image signature generation and embedding. Compared to JPEG, JPEG2000[4][5] is a new image compression standard, which adopts wavelet transform (WT) for better energy compaction, and provides greater flexibility in scalability, bit stream ordering, and decoding. One of the objectives of this paper is to develop content-based image features robust against JPEG2000 compression but sensitive to content altering attacks.

First, we consider some JPEG2000 targeting applications and identify a list of acceptable manipulations, which may not be complete.

- Recompression—Typically this involves re-encoding a compressed image to a lower bit rate. The re-encoding may be repeated for multiple times. In practice, there is a minimal requirement of image quality; thus, one may set a minimal acceptable bit rate beyond which the image will be considered as unauthentic.
- Format / Codec Variations – Difference may exist among different implementations of JPEG2000 codec by different companies. Such differences could be due to different accuracies of representation in the domains of (quantized) WT, color transformation, and the pixel domain.
- Watermarking – Image data is "manipulated" when authentication feature codes are embedded back to the image. Such manipulation should not make the result image become un-authentic.

In [1][2], we have proposed a generic semi-fragile image authentication framework combining ECC and PKI. The framework is general and can be used to embed any specific feature codes. The content signing procedure includes signature generation and watermark embedding while the content authentication procedure includes watermark extraction and signature verification. The use of ECC addresses incidental distortions caused by acceptable manipulations such as lossy compression. Integration of PKI security infrastructure and a subsequent hashing mechanism achieves system security and signature length reduction. In the signing procedure, block-based invariant features are extracted from the image content and then encoded by ECC to derive corresponding parity check bits (PCB). All PCBs are then embedded back into image as watermarks for content authentication and alteration location. As another layer of authenticity checking, original feature codes from each block are concatenated to form a global signature, hashed for length reduction, and finally signed by content owner's private key for cryptographic protection.

In this paper, we apply the above framework to JPEG2000 and describe a solution in extracting robust content-based features from JPEG2000 specific processes. More specifically, we describe how to quantitatively extract invariant feature from the EBCOT process and study their invariant properties under various acceptable manipulations. One unique concept we use in the system is *lowest authenticable bit rate (LABR)*. One objective of the system is to make all JPEG2000 compressed images pass the authenticity verification process provided the compression rate is not less than the pre-determined authentication bit rate. For this, our system explores the synergistic rate-distortion control mechanism provided in the EBCOT process in JPEG2000. Details of the approach are described later. Note in [3], a related concept was proposed to authenticate images compressed at a quantization level no larger than a pre-set level. However,

for content owners, setting quantization levels is less intuitive than directly setting the allowed bit rate.

The rest of the paper is organized as follows. In Section 2, we discuss the overall system combining our ECC-PKI authentication framework with the invariant features, derived from JPEG2000 processes. In Section 3, we shall discuss the invariant features for JPEG2000 authentication. Some experimental results are shown in Section 4, followed by conclusions and future work in Section 5.

2. SYSTEM OVERVIEW

In our proposed solution, signature generation / verification modules are mainly employed for the role of content signing and authentication. Watermark embedding / extraction modules are only used for storing signatures. As shown in Figure 1, the procedure of content signing can be described as follows. Three inputs are needed to sign a JPEG2000 image: original image, image owner's private key and a parameter specifying the LABR. If a JPEG2000 image is signed with LABR value *a*, a new image with bit rate *b* will be authentic as long as *b* is greater than *a* and the new image is derived from defined acceptable manipulations. The input original image first goes through color transformation, wavelet transformation and quantization, which are all basic procedures in JPEG2000 encoding. It is then partitioned into non-overlapping blocks (called codeblock) in each resolution level and subband. EBCOT [6] (embedded block coding with optimized truncation) is employed for bit plane encoding and optimal bit rate control. EBCOT separate the wavelet coefficients in a codeblock into more truncation layers than bit planes by using multi-pass bit plane fractionalization and context-model based bit ordering. Figure 2 shows passes (vertical axis) of each codeblock (horizontal axis) at different encoding target rate. We extract content-based features from the available fractionalized bit planes when the image is encoded above LABR. Details of the content-based features are described in the next section. Feature values from each code block are ECC coded to generate corresponding parity check bits (PCBs). Then we take PCBs as the seed of watermark to form the block-based watermark. One necessary condition in watermarking is that the embedded watermark should be robust enough for extraction from received images under acceptable manipulations. Since incidental changes to the embedded watermarks might occur, we apply ECC again before PCB data is embedded. The watermark data for each block is embedded either into the same block or into a different block. The watermark embedding location is also determined based on the LABR value. We focus on images that will not be coded or transcoded to a bit rate lower than LABR. From the EBCOT rate control mechanism, we know exactly the specific passes that will always be available as long as the image is encoded at a rate above LABR. EBOCT also provides a convenient mechanism for computing the distortion when watermarks are embedded into selected passes of the code block. Such mechanism allows us to control the quality degradation caused by watermark embedding. The simplest way of embedding process is to replace the selected bit plane with the bits from the embedded signature. Human visual models can also be adopted to adaptively select the blocks that are more amenable to embedding.

Note only the PCB data (not including the feature codes) are embedded in the above watermarking process. Although the feature codes are generated block by block and thus provide localization capability, the embedded data may be changed under acceptable manipulations. To overcome this problem, we add another signature generated in a global fashion. All codewords (features together with their corresponding PCBs) are concatenated and the resulting bit sequence is hashed by a cryptographic hashing function such as MD5[7]. Finally, content owner's private key is used to sign the hashed value. The encrypted hashed value can be stored in a place external to the image content such as its file header or embedded into the image again as another watermark.

The authentication procedure is the inverse procedure of signing except using content owner's public key for signature verification. Given the LABR, we repeat feature extraction for each codeblock. From the embedded watermarks, we extract PCB data generated at the source site. Note the features are computed from the received image, while the PCBs are recovered from the watermarks that are generated and embedded at the source site. After we combine the features and the corresponding PCBs to form codewords, the whole authentication decision could be made orderly. First, we calculate the syndrome block by block to see whether there exists any blocks whose codewords are uncorrectable. If yes, then we claim the image is unauthentic and use the above ECC checking process to find alteration locations. If all codewords are correctable (i.e. errors in any feature code are correctable by the PCB), we repeat the same process as the source site: concatenate all corrected codewords into a global sequence and cryptographically hash the result sequence. By using owner's public key, the authenticator can decrypt the hashed value that's generated at the source site. The final authentication result is then concluded by bit-by-bit comparison between these two hashed sets: if there is any single bit difference, the authenticator will report that the image unacceptable ("unauthentic"). Note there exists chance that every code block passes the checking process but the global hashed signatures do not agree.

3. CONTENT-BASED FEATURE EXTRACTION

We select EBCOT to generate robust content-based features based on following considerations. First, EBCOT is the last processing unit prior to forming the final compressed bitstream in JPEG2000. It means all possible

distortions have been introduced before running EBCOT in the encoding procedure while no distortion is introduced afterwards. If we are to authenticate the encoding output or output of directly truncating the compressed bitstream, the extracted features will not be distorted. Secondly, EBCOT expands the scalability of image by fractionalizing the bit-plane. The significant bit planes of EBCOT represent closely the image content and thus it is hard to alter the image while intentionally keep the same significant bit planes. Thirdly, in JPEG2000 EBCOT is the engine of bit rate control and provides exact information about specific passes of data to be included in the final bit stream given a target bit rate. Such information allow us to specify the invariant layers of data and quantitatively select an authenticable level.

EBCOT provides a finer scalability of image content resulted from multiple pass encoding on the codeblock bit-planes (i.e., fractionalizing each bit-plane into 3 sub-bit-planes: significant pass, magnitude refinement pass and clean-up pass based on pre-defined contextual models). The coding summaries for each codeblock after EBCOT include feasible truncation points, their corresponding compressed size (rate), and estimated distortions (rate-distortion curve). The target compression bit rate is achieved by globally scanning the contributions from all codeblocks and optimally selecting truncating points for each codeblock (using Lagrange Multiplier method). One important property is worth noting – passes included in the bit stream at a rate (say a) are always included in the bit stream at a higher rate ($>=a$). Such property is illustrated in Figure 2, in which no curves cross the curves correspond to passes included at different compression bit rates. Such property is important for our proposed solution in order to obtain invariant feature sets.

Based on above observation, we select two measures as invariant features directly from EBCOT: one is the state of passes (i.e., the fractionalized bit planes) of MSBs and the other is the estimated distortion associated with each pass. The estimated distortion is a measure of the change of "1" in a given bit-plane (details in [5]). Their invariant properties under acceptable manipulations are shown in Figure 3 and 4 respectively. Figure 3 shows an example of differences among significant passes (SP) of one code block from different codec implementations. The test software are obtained from [8][9] where JO means SP of original image based on an encoder implemented in Java, CO means the difference between SP from two different encoders (Java vs. C++), JOC and COC mean the differences between JO and SP after multiple compression cycles (Java-based and C++ based implementations) respectively, JA and CA mean the differences between JO and SP of attacked image (Java-based and C++ based respectively), JAC and CAC mean the differences between JO and SP of attacked image plus multiple compression cycles (Java-based and C++ based respectively). As we expect, codec implementation variations and acceptable re-compression introduced much less changes to the bit plane than content-altering attack. Similarly, as shown in Figure 4, estimated distortions associated with the significant passes of EBCOT are much more stable during acceptable manipulations than attacks. However, as seen from Figure 3 and 4, there are still some small perturbations introduced to the features. This is the reason why we employ ECC to tame such distortions in order to generate stable signatures.

4. EXPERIMENTAL RESULTS

We tested the proposed methods under different compression bit-rates and Lowest Authenticable Bit Rates. The averaged PSNR between coded JPEG2000 images and coded plus watermarked JPEG2000 image is shown in Figure 5 under different bit-rates. It shows watermarking causes only slight degradation (relative to the original SNR values). We select 3 bit-rates for our tests: full bit-rate (i.e., quantization step size is minimal and no any truncation on formed bitstream and its actual compression bit-rate is around 2.5-4bpp and LABR: 2bpp), 1bpp (LABR: 0.8bpp) and 0.25bpp (LABR: 0.2bpp). In a sub-codeblock with size 16x16, the generated feature length is around 262 from two features (256 from SP and 7 from distortion). After ECC, the length of PCBs is around 24 (16 bits for SP and 8 bits for distortion) if the selected ECC schemes are BCH (255, 239, 2) and (15, 7, 2) respectively. The length of generated watermarks is 46 bits (31 for SP and 15 for distortion) if the ECC schemes for watermarking are BCH (31,16,3) and (15, 7, 2). The watermarking scheme we select for our testing is a modified version of [10]. We further decode and re-encode the watermarked images 1, 5, and 10 times under different transcoding bit-rates. We achieve successful authentication (over various acceptable manipulations mentioned above) in all cases, except very few blocks. More details about testing results, distortion analysis and watermarking issues will be given in future publications.

Note JPEG2000 provides different scalability modes, among which SNR scalability and resolution scalability are two most well known. Different scalable encoding data can be packed in the final bit stream following different progression orders (e.g., SNR first followed by resolution). If we restrict the transcoding operations to be the one set at the encoder (e.g., drop the SNR layers only, or drop resolution layers only), we know exactly what code blocks and layers that will not change during transcoding, and thus we can use the unchanged layers for feature extraction. If the transcoding process does not follow the restriction, only significant layers in coarse wavelet levels can be used. This is a tradeoff between application flexibility and performance (in terms of sensitivity in detecting attacks).

5. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a novel solution for semi-fragile authentication of JPEG2000 images. Our system used

invariant content-based features derived from JPEG2000 specific domains. Specifically, we derived stable features from the EBCOT process. They include the fractionalized bit planes and estimated distortions of significant bits of JPEG2000 code blocks. Although limited, our experiments showed the effectiveness of such features in distinguishing the acceptable manipulations (such as JPEG2000 compression) and content altering attacks. For acceptable lossy compressions, we further consider the effects caused by variations in codec implementations and transcoding modes. To enhance security and efficiency of the system, we apply our previous proposed framework to combine the content-based signature with PKI and ECC infrastructures. The whole procedure of signature gene-ration / verification and watermark embedding / extraction can be efficiently incorporated into JPEG2000 coding scheme.

Future work will focus on increasing system robustness to tolerate more acceptable manipulations such as image filtering/sharpening by using multiple signatures and watermarks in other domain such as WT coefficients. Tradeoffs between robustness and embedding capacity will studied as well.

5. REFERENCES

[1] Qibin Sun and Shih-Fu Chang, Semi-Fragile Authentication of JPEG-2000 Images with a Bit Rate Control, *Columbia University ADVENT Technical Report*, 2002-101.

[2] Qibin Sun, Shih-Fu Chang, Maeno Kurato and Masayuki Suto, "A new semi-fragile image authentication framework combining ECC and PKI infrastructure", *ISCAS02*, Phoenix, USA, May, 2002.

[3] Ching-Yung Lin and Shih-Fu Chang, "Semi-Fragile Watermarking for Authenticating JPEG Visual Content," *SPIE Security and Watermarking of Multimedia Contents II EI '00*, SanJose, CA, Jan. 2000.

[4] Information Technology—JPEG2000 Image Coding System, *ISO/IEC International Standard 15444-1*, ITU Recommendation T.800, 2000.

[5] M. Rabbani and R. Joshi, An overview of the JPEG2000 still image compression standard, *Signal Processing: Image Communication*, Vol.17, No.1, 2001.

[6] D. Taubman, High performance scalable image compression with EBCOT, *IEEE Transactions on Image Processing*, Vol.9, No.7, pp.1158-1170, Jul. 2000.

[7] B. Schneier, *Applied Cryptography*, New York: Wiley, 1996.

[8] Joint Photographic Experts Group: www.jpeg.org.

[9] JJ2000: An implementation of JPEG2000 in JAVA™, available at http://jj2000.epfl.ch.

[10] M. Wu, E. Tang and Bede Liu, Data hiding in digital binary image, *Proceedings of ICME'00*, 2000

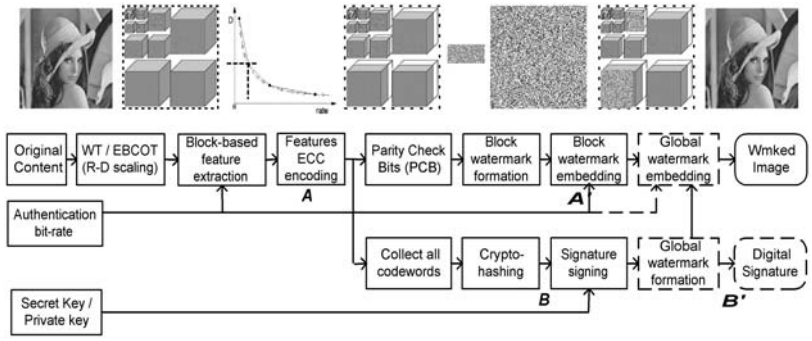

Figure 1. Proposed semi-fragile JPEG2000 image authentication system

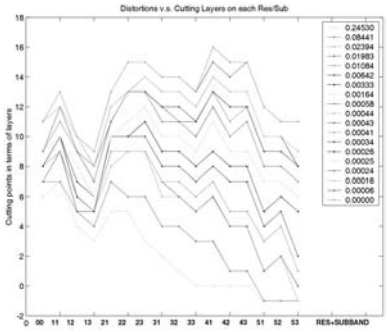

Figure 2. An example of all Cutting layers

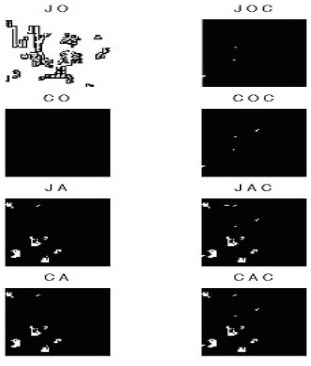

Figure 3. The Significant pass under attacks and variations in codec implementations and re-compression.

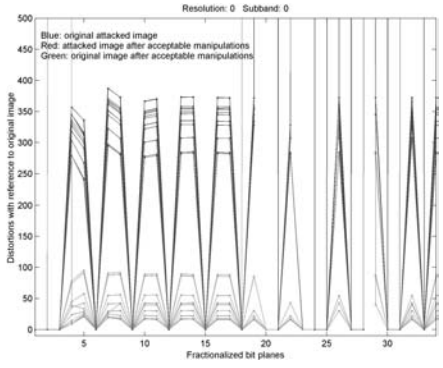

Figure 4. Changes in estimated distortions associated with significant passes. The x axis Indicates the SP index while the y axis shows the estimated distortion.

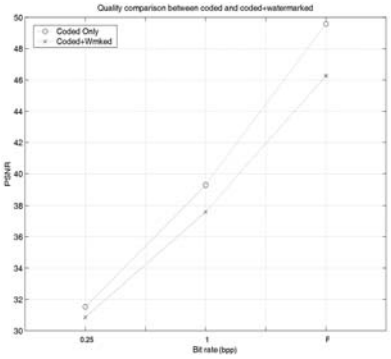

Figure 5. Image SNR values (top curve: JPEG200 compression only bottom curve: compression plus watermarking)