PROGRESSIVE PROTECTION OF JPEG2000 CODESTREAMS

Yongdong Wu, Di Ma and Robert H. Deng

Institute for Infocomm Research
21, Heng Mui Keng Terrace, Singapore, 119613
{wydong, madi, deng}@i2r.a-star.edu.sg

ABSTRACT

This paper presents an authentication scheme and an access control scheme for JPEG2000 image codestreams based on hash chains. Both schemes are fully compatible with the core part of JPEG2000 standard. The authentication scheme allows users to verify the authenticity of sub-images progressively extracted from a single codestream protected with a single signature, and the access control scheme allows users to access sub-images partially decrypted from a single encrypted JPEG2000 codestream. In addition, the two schemes can be integrated to provide both authenticity and access control simultaneously. Our experiments demonstrate the two scheme are very efficient and highly scalable.

1. INTRODUCTION

JPEG2000 [1]-[3] is the latest international still image compression standard. One of its remarkable merits is "compress once, decompress many ways", *i.e.*, it supports extraction of transcoded images of different resolutions, quality layers and regions, all from a single compressed codestream. This merit allows applications to manipulate or disclose only the required image data of a codestream based on users' privileges or capabilities. JPEG2000 standard document Part 8, named JPSEC, is work in progress and is concerned with all the security aspects of JPEG2000 image codestreams and files.

Grosbois et al [4] proposed an authentication scheme using digital signature for JPEG2000 codestreams. The scheme simply digitally signs each code-block individually and then attaches the digital signature to the end of the code-block bit stream. Hence, the scheme is inefficient and vulnerable to *collage attacks* which replace or reorder signed code-blocks. Grosbois et al [4] also presented an access control method that allows a preview of low resolutions of an image, whilst prevents the correct display of higher resolutions by introducing pseudo-random noise in high frequency subbands. This access control method is simple but, unfortunately, is vulnerable to the so called *known plaintext attacks*.

Wu et al [5][6] presented an authentication scheme for JPEG2000 image code-steams employing the *Merkle hash tree* and a digital signature algorithm. The scheme is fully compatible with JPEG2000's "compress once, decompress many ways" functionality and possesses a "sign once, verify many ways" property. That is, the scheme allows users to verify the authenticity of different sub-images extracted from a single compressed codestream protected with a single digital signature. This scheme is very flexible at the price of moderately increased payload.

The Secure Scalable Streaming (SSS) technique proposed in [7] supports low complexity, high quality transcoding at intermediate, possibly untrusted, network nodes without compromising the end-to-end security of the system. SSS encodes, encrypts, and packetizes video into secure scalable packets in a manner that allows transcoders to perform transcoding operations (e.g., bit rate reduction and spatial down sampling) by simply truncating or discarding packets, and without decrypting the data. Secure scalable packets have unencrypted headers that provide side information, such as optimal truncation points, to downstream transcoders. However, key management, the important topic in our paper, is not considered in [7].

This paper presents an authentication scheme and an access control scheme for JPEG2000 image codestreams. Both schemes are fully compatible with the core part of the JPEG2000 standard. The authentication scheme allows users to verify the authenticity of any transcoded sub-images extracted from a single codestream protected with a single signature. The access control scheme allows users to access truncated codestreams from a single encrypted JPEG2000 codestream.

Although our schemes are designed independently, they can be merged into a composite scheme so as to provide progressively authentic access to JPEG2000 codestreams. One approach to achieve this is to protect packets in a codestream with the access control scheme first, then the access controlled packets are further protected using the authentication scheme. Nevertheless, the decryption key for the truncated codestream in the composite scheme should be authentic because the authentication scheme ensures the authenticity of the protected codestream, other than the decrypted codestream. Fortunately, it is easy to detect a cor-

0-7803-8554-3/04/$20.00

rupted decryption key since a codestream decrypted with a wrong key results in noise, which in turn can be detected easily by performing a correlation analysis.

The rest of this paper is organized as follows. Section 2 introduces some basic concepts in JPEG2000, those who are familiar with JPEG2000 can skip this section. Section 3 describes our progressive authentication scheme and Section 4 presents the progressive access control scheme. Our experiment results are shown in Section 5. A conclusion is drawn in Section 6.

2. OVERVIEW OF JPEG2000 CODESTREAMS

According to [1]-[3], a JPEG2000 image codestream is organized hierarchically with several types of structural elements - tiles, components, tile-components, resolution levels, precincts, layers and packets.

Tiles: an image is partitioned into several tiles and each tile is treated as a smaller independent image for the purpose of compression.

Components and Tile-components: A component refers to an element, such as Red, Green, and Blue.

Resolution: Given a tile-component, a multiple-level DWT is performed. A wavelet transform decomposes a tile - component into four sub-bands. Applying the wavelet transform continuously on each lowest frequency sub-band generates a series of sub-bands belonging to different transform levels. A $(n_r - 1)$-level wavelet transform generates $3n_r - 2$ sub-bands which construct n_r sub-images at resolution 0, resolution 1, ..., and resolution $n_r - 1$, respectively.

Precincts: Precincts are used to make it easier to access the wavelet coefficients corresponding to a particular spatial region of the image.

Layers and Quality levels: Assuming that a tile is partitioned into n_l layers. The sub-image with quality level 0 has the lowest quality, and the sub-image with quality level $n_l - 1$ is the original image which has the highest quality.

Packets: Packets are the most fundamental building blocks of JPEG2000 codestreams. A JPEG2000 codestream can be viewed as a set of packets. In a codestream, a packet in a given tile is uniquely identified by four parameters: C (component), R (resolution level), P (precinct) and L (layer). The packets of an image codestream are sorted with respect to these four parameters in some orders, called progression orders. JPEG2000 allows five progression orders - LRCP, RLCP, RPCL, PCRL and CPRL.

3. PROGRESSIVE AUTHENTICATION

Figure 1 illustrates the application (e.g. in government and court of law) of the proposed progressive authentication scheme. It includes three fundamental operations, authentic codestream generation, truncation and verification which are carried out at a producer, a proxy and a user, respectively.

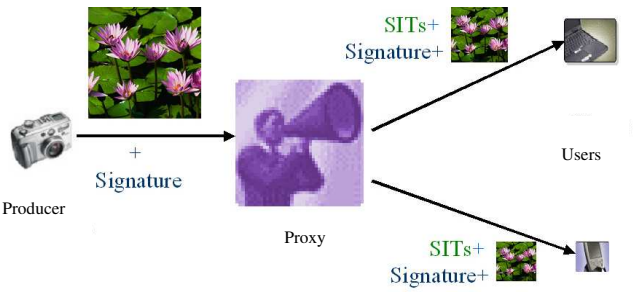

Fig. 1. An application scenario of the progressive authentication scheme where SIT stands for Subsidiary Integrity Token.

3.1. Generating Progressive Authentic Codestream

After an image is captured, the image is compressed using a standard JPEG2000 encoder to produce a JPEG2000 codestream of the image . We assume that there are *n* packets in the JPEG2000 codestream, denoted as P_i, *i* = 1,2,. . ., *n*. To generate the authentic codestream, the producer calculates incrementally hash values of the *n* packets as follows,

$$h_n = H(P_n) \quad (1)$$

$$h_i = H(P_i||h_{i+1}) \qquad i = n-1, \ldots, 2, 1 \quad (2)$$

where $H(\cdot)$ is a one way hash function, and "||" is the concatenation operator. Note that the tile header and/or codestream header is treated as part of the packet immediately following it when performing the hash operations in Equations (1) and (2). The producer signs on the hash value h_1 to generate a signature σ for the codestream using a underlying signature scheme such as RSA. The authentic codestream consists of σ, P_i, *i* = 1,2,. . ., *n* and is forwarded to the proxy for dissemination to users.

3.2. Truncating Authentic Codestream

Due to constraint of network bandwidth or user devices, the proxy may have to discard part of the codestream. Using JPEG2000's EBCOT property [1]-[3], the truncation of the codestream starts from packets with high index values. For example, to discard $n - n_0 + 1$ packets, the proxy discards the packets $P_i (i = n_0, n_0+1, \ldots, n)$, computes the SIT h_{n_0} using Equations (1) and (2). The proxy then sends the truncated codestream consisting of packets P_i $(i = 1, 2, \ldots, n_0 - 1)$, the signature σ and the SIT h_{n_0} to the user.

3.3. Verifying Truncated Codestream

To verify the truncated codestream P_i $(i = 1, 2, \ldots, n_0-1)$, the user calculates

$$\begin{aligned} h_{n_0-1} &= H(P_{n_0-1}||h_{n_0}) \\ h_i &= H(P_i||h_i+1) \qquad i = n_0-2, \ldots, 2, 1 \end{aligned}$$

Then, the user checks h_1 against σ with the public key of the producer. If h_1 matches σ, the truncated codestream is considered authentic.

3.4. Performance

The proposed progressive authentication scheme allows the proxy to flexibly truncate an authentic codestream according the constraint of the network bandwidth or user device capability. The user can verify the authenticity of the truncated codestream using the SIT and the producer's signature. Comparing with the naïve solution in which the producer or the proxy signs on the truncated codestream each time, our scheme is computationally efficient (since hash computation is much more efficient than signature generation) and secure (since the producer's signature private key is not kept by the proxy). The transmission overhead in our scheme is also very small, only a hash value and a digital signature are transmitted in order to allow the user to verify the authenticity of the truncated codestream.

4. PROGRESSIVE ACCESS CONTROL

In this section, we propose an access control method which manages the access control keys using a hash chain, where a node of the hash chain is an encryption key which is used to encrypt a corresponding packet in a codestream. With respect to Figure 2, the first key in the chain corresponds to the last packet of the codestream, the second key corresponds to second last packet, and so on. The encrypted codestream is distributed freely. To access truncated codestreams, a user interacts with an on-line key server to obtain appropriate keys. Only one key is needed by the user to decrypt and access a truncated codestream.

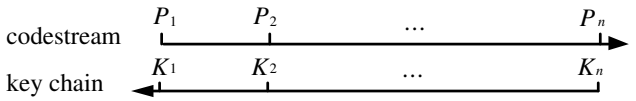

Fig. 2. Scenario of progressive access. In this scheme, the progressive order of the packets is different from that of the keys.

4.1. Generating Progressive Access Key by the Producer

Again we assume that there are *n* packets in a JPEG2000 codestream, $P_i, i = 1, 2, \ldots, n$. Then the producer create a hash chain with a master key *K*

$$K_i = H^{n-i}(K) = H(H^{n-i-1}(K)) \;\; i = n, \ldots, 2, 1$$

where $H(\cdot)$ is a one way hash function, $H^0(K) = K$. Then the producer encrypts the packet P_i with the key K_i, $i = 1, 2, \ldots, n$, and distributes the encrypted codestream freely to users. Note that delimiting markers in packets are not encrypted. This is important in order to have correction decryption of the encrypted packets by users.

4.2. Rendering the Requested Codestream by the User

Assuming that a user desires to access a truncated codestream consisting of packets $P_i, i = 1, 2, \ldots, p$, the user authenticates himself or herself (or makes appropriate payment) to an on-line key sever. The server in turn sends the key K_p to the user. The user computes

$$K_i = H^{p-i}(K_p) \qquad i = p-1, \ldots, 2, 1 \qquad (3)$$

Then, the user decrypts packet P_i with key K_i, *i* = 1,2, . . ., *p* and obtains the desired truncated codestream P_i, *i* = 1,2, . . ., *p*.

4.3. Performance

In our access control scheme, the producer prepares the encrypted codestream only once. The encrypted codestream can be distributed freely, either on-line or off-line. To access a truncated codestream, only a single key is sent from the on-line key server to a user. Therefore, the proposed scheme is extremely efficient in key management. Obviously, the security strength of the scheme is the same as the underlying hash function and encryption function.

5. EXPERIMENTS

To simplify the description, we show experiment results of the two proposed schemes independently. The application of the two schemes simultaneously to a codestream is straightforward and is not discussed here.

5.1. Progressive Authentication

In our experiment, the example image codestream consists of 3 layers and 3 resolutions, and the digital signature scheme is a 1024-bit RSA.

To test the authenticity of the truncated codestream by layers, we first digitally sign the example codestream with packets arranged in progressive order LRCP. The truncated images are shown in Figure 3.

Fig. 3. Progressive authenticity by layers for LRCP ordered codestream, where the leftmost image is of the lowest quality, and the rightmost is of the highest quality.

Subsequently, the original codestream is rearranged in RLCP progressive order and is signed again. The codestream is truncated by resolution as shown in Figure 4.

Fig. 4. Progressive authenticity by resolutions for RLCP ordered codestream, where the leftmost image is of the lowest resolution, and the rightmost is of the highest resolution.

All the images in Figure 3 and Figure 4 are verified to be authentic. To demonstrate the detection of tampering, we modify the codestream randomly, the tampered image is shown in Figure 5, the scheme reports that the image shown in Figure 5 is tampered.

Fig. 5. A tampered image. The tampering in the top-right corner is detected by our scheme.

5.2. Progressive Access Control

In this experiment, the example codestream comprises of 5 resolutions. To test the progressive access, packets of the codestream are arranged in RLCP progressive order and are encrypted according to the method in Section 4, where each packet is encrypted with a unique key. To decrypt an image of specific resolution, the on-line key server releases the key for the last packet of the required resolution. The user calculates the other keys one by one according to Equation (3) and then decrypts the corresponding packets. In Figure 6, the images with different resolutions are shown.

Fig. 6. Progressive access. The left image is decrypted with the key for resolution 3, and the right one is decrypted with the key for resolution 4. However, it is impossible to obtain the right image using the key for resolution 3.

6. CONCLUSION

JPEG2000 is an emerging international standard for image compression. JPEG2000 security (JPSEC) is in progress and is concerned with JPEG2000 codestream security with particular emphasis on flexible authentication and access control. To meet the requirements of JPSEC, schemes proposed in this paper use hash chains so as to exploit the data structure of JPEG2000 codestreams. Our experiments demonstrated that the proposed schemes are scalable and efficient.

7. REFERENCES

[1] D. S. Taubman and M. W. Marcellin, *JPEG2000 - Image Compression Fundamentals, Standards and Practice*, Kluwer Academic Publishers, 2001.

[2] M. Rabbani and R. Joshi, "An overview of the JPEG 2000 still image compression standard", *Signal Processing: Image Communication*, 17(1):3-48, 2002.

[3] "Information technology - JPEG 2000 image coding system", *ISO/IEC International Standard 15444-1*, ITU Recommendation T.800, 2000

[4] R. Grosbois, P. Gerbelot and T. Ebrahimi, "Authentication and Access Control in the JPEG 2000 Compressed Domain", SPIE Vol. 4472, pp. 95-104, 2001.

[5] Yongdong Wu, Robert Deng, Di Ma, Cheng Peng, and Yanjiang Yang, "Authentication of JPEG 2000 Codestreams and Files", ISO/IEC JTC 1/SC 29/WG1/N2809, Mar. 2003

[6] Cheng Peng, Robert Deng, Yongdong Wu, Weizhong Shao, "A Flexible and Scalable Authentication Scheme for JPEG2000 codestreams," ACM Multimedia, pp.433-441, Nov. 2003

[7] S. Wee and J. Apostolopoulos, "Secure Scalable Streaming and Secure Transcoding with JPEG-2000," IEEE ICIP 2003