A Key Derivation Scheme for Hierarchical Access Control to JPEG 2000 Coded Images

Shoko Imaizumi[1], Masaaki Fujiyoshi[2], Hitoshi Kiya[2], Naokazu Aoki[1], and Hiroyuki Kobayashi[1]

[1] Graduate School of Advanced Integration Science, Chiba University, 1–33 Yayoicho, Inage-ku, Chiba-shi, Chiba, Japan
[2] Dept. of Information and Communication Systems, Tokyo Metropolitan University, 6-6 Asahigaoka, Hino-shi, Tokyo, Japan
imaizumi@chiba-u.jp, fujiyoshi-masaaki@tmu.ac.jp, kiya@sd.tmu.ac.jp, {aoki,kobahiro}@faculty.chiba-u.jp
http://www.nd.chiba-u.jp/yugo-index_e.html

Abstract. This paper proposes a key derivation scheme to control access of JPEG 2000 (JP2) coded images, which consist of hierarchical scalability such as SNR, resolution levels, and so on. The proposed scheme simultaneously controls access to each level of scalability. The proposed scheme derives keys through hash chains, and each JP2 packet is enciphered with each individual key. By introducing combinations of a cyclic shift and a hash function, the proposed scheme manages only a single key for a JP2 image; whereas the conventional access control schemes having the above mentioned features manage multiple keys. The single managed key is not delivered to any user. The proposed scheme is also resilient to collusion attacks. Performance analysis shows the effectiveness of the proposed scheme.

Keywords: JPEG 2000, access control, key derivation, hash chain, cyclic shift.

1 Introduction

With the continuing growth in communication channels and terminals, scalable transmission, in which lower quality content is displayed by decompressing a certain portion of the codestream, is becoming popular. Scalable access control for the protection of scalable compressed images has been studied widely [1,2,3, 4,5,6,7,8,9,10]. Security for JPEG 2000 (JP2) [11] is emphasized in JPEG 2000 Part8 [12], and JP2 coded images must be tightly secured.

A simple and straightforward way to realize hierarchical access control for JP2 coded images, consisting of several kinds of scalability, is the individual encipherment of each JP2 packet. This approach, however, must manage a large number of keys, given the large number of JP2 packets in a JP2 coded image.

Scalable access control schemes have also been proposed for JP2 coded images [3, 4, 5, 6, 7, 8, 9, 10]. These schemes use one- or multi-dimensionally hierarchical scalability provided by coding technologies, so that the user can obtain

Y.-S. Ho (Ed.): PSIVT 2011, Part II, LNCS 7088, pp. 180–191, 2011.

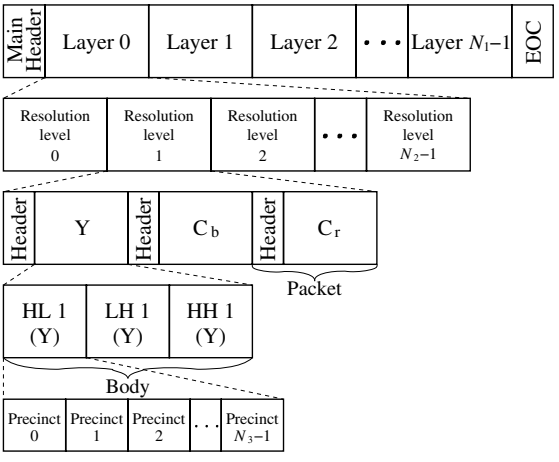


Fig. 1. JP2 codestream with color components, Y, C_b, and C_r. The progression order is LRCP.

an image or a video at the permitted quality from one common codestream. The hash chain [13] has also been introduced to several schemes for reduction of the number of managed keys and the keys delivered to the user (delivered keys) [6, 7, 8, 9, 10]. Although these hash chain-based access control schemes are effective for reduction of the number of keys, the number of managed keys increases, depending not only on the kinds of scalability, but also on the depth of the hierarchy in each scalability.

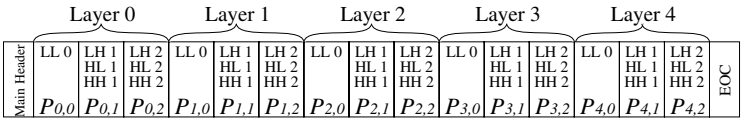
This paper proposes an efficient key derivation scheme for hierarchical access control to JP2 coded images in which several kinds of scalability exist. By introducing combinations of a cyclic shift and a hash function, the number of managed keys is reduced to one. The managed key is not delivered to any user, providing security against key leakage. The proposed scheme is also resilient to collusion attacks, in which malicious users illegally access an image at higher quality than that allowed by their access rights.

2 JP2 Codestream and Hierarchical Access Control

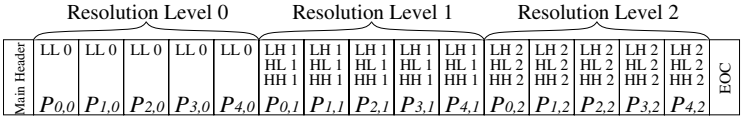
This section briefly describes JP2 codestream structure [11] and scalable access control for JP2 coded images. It also summarizes the requirements for hierarchical access control methods by introducing four conventional methods [7, 8, 9, 10] to clarify the aim of this work.

2.1 JP2 Codestream

Fig. 1 outlines a JP2 codestream using YC_bC_r as the color space. JP2 supports five different progression orders that are orders of scalability, and the default order, that is also used in Fig. 1, is LRCP (Layer-Resolution-Component-Precinct). It is primarily progressive by quality.



(a) LRCP.



(b) RLCP.

Fig. 2. Progression orders in a grayscale image with five layers and three resolution levels, i.e., $N_1 = 5$ and $N_2 = 3$

Layers are in order of SNR in which each layer is composed of data for resolution levels. If the original image has color components, each resolution level has Y, C_b, and C_r components. Resolution level zero only contains the LL data, whereas the other levels contain three subbands; HL, LH, and HH. These subbands have precincts that have non-hierarchically positional information. Thus, a color JP2 codestream has three kinds of hierarchical scalability; layer, resolution revel, and components, whereas a grayscale JP2 codestream has two; layer and resolution level. Each JP2 packet is composed of a header and a body and contains partial data for each subband.

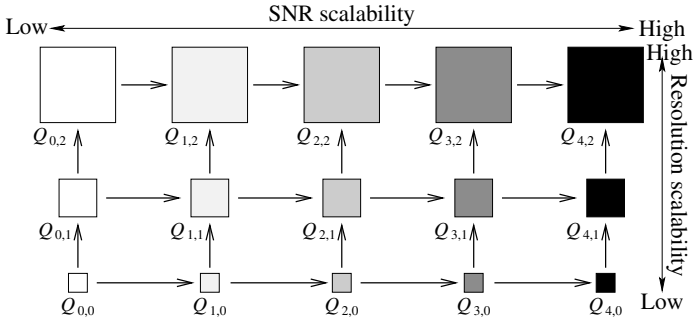
Fig. 2 lists examples of JP2 codestreams with LRCP and RLCP progression orders. Both have five layers and three resolution levels, which are represented as $N_1 = 5$ and $N_2 = 3$, respectively, in this paper. Hereafter, P_{n_1,n_2} is the JP2 packet at the n_1-th layer and n_2-th resolution level.

2.2 Hierarchically Access Control

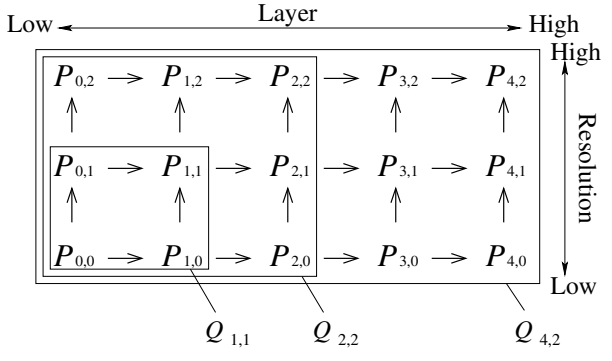
Fig. 3 outlines an example of scalable decoding in which different image products are obtained by decompression in many ways, where $N_1 = 5$ and $N_2 = 3$. In this example, the decoded image is grayscale. It is noted that this representation holds regardless of progression orders. The original image is compressed at quality $Q_{4,2}$, and the image at $Q_{4,2}$ is obtained by decompressing all packets. To produce the image at $Q_{1,1}$, four packets $P_{0,0}$, $P_{0,1}$, $P_{1,0}$, and $P_{1,1}$ are decompressed. Thus, a scalable access control method for JP2 should encipher a JP2 codestream packet-by-packet using $N_1 \times N_2$ different keys. Access control for JP2 coded images should encipher the packet body but does not encipher the packet headers.

2.3 Requirements

This section describes two requirements for hierarchical access control for JP2 coded images, i.e., collusion attack-resilience and the less number of managed keys.



(a) Quality at Q_{n_1,n_2}.

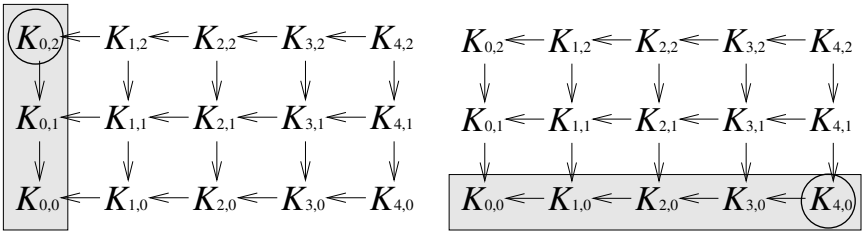


(b) Decompression to P_{n_1,n_2}.

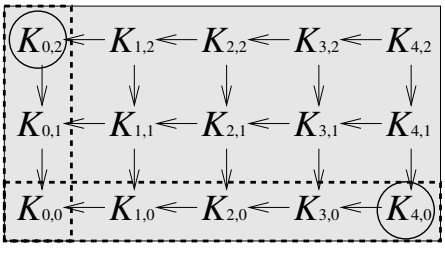
Fig. 3. Hierarchical decompression of a grayscale image ($N_1 = 5$ and $N_2 = 3$)

Collusion Attack-Resilience. A collusion attack is made by multiple users to obtain an image with higher quality than that allowed by their access rights, and the conventional scheme [7] allows users to collude. In Fig. 4, the arrows indicate direction of key derivation. The key for packet P_{n_1,n_2} is K_{n_1,n_2}, and $K_{4,2}$ is the managed key. As shown in Fig. 5, the managed key $K_{4,2}$ is divided into two partial keys $K_{\mathrm{PK}_1(4)}$ and $K_{\mathrm{PK}_2(2)}$. Each partial key is allocated to each hierarchy, and the partial keys $K_{\mathrm{PK}_1(n_1)}$ and $K_{\mathrm{PK}_2(n_2)}$ for key K_{n_1,n_2} are derived from previous partial keys $K_{\mathrm{PK}_1(n_1+1)}$ and $K_{\mathrm{PK}_2(n_2+1)}$, using hash chains [13]. By concatenating them, $K_{n_1,n_2} = \left(K_{\mathrm{PK}_1(n_1)} \parallel K_{\mathrm{PK}_2(n_2)}\right)$, is derived.

In Fig. 4 (a), Alice is allowed to access the image at $Q_{0,2}$ and receives key $K_{0,2}$, which is consisting of two partial keys $K_{\mathrm{PK}_1(0)}$ and $K_{\mathrm{PK}_2(2)}$. She can derived keys $K_{0,1}$ and $K_{0,0}$ and decipher $P_{0,2}$, $P_{0,1}$, and $P_{0,0}$. Whereas, Bob, in Fig. 4 (b), receives $K_{4,0}$, which is consisting of $K_{\mathrm{PK}_1(4)}$ and $K_{\mathrm{PK}_2(0)}$, and derives $K_{3,0}$, $K_{2,0}$, $K_{1,0}$, and $K_{0,0}$ to decipher $P_{4,0}$, $P_{3,0}$, $P_{2,0}$, $P_{1,0}$, and $P_{0,0}$ for access the image at $Q_{4,0}$. In this scheme, they are possible to illegally derive $K_{4,2}$ by using $K_{\mathrm{PK}_1(4)}$ and $K_{\mathrm{PK}_2(2)}$, so they can decipher all packets as shown in Fig. 4 (c) and access the image at $Q_{4,2}$. The proposed scheme is resistant to collusion attacks.



(a) Keys legally obtained by Alice. (b) Keys legally obtained by Bob.



(c) Keys obtained by a collusion attack.

Fig. 4. Alice and Bob's collusion attack in the vulnerable scheme [7] (the shaded are obtained)

The Less Number of Managed Keys. Although a hierarchical access control method requires $N_1 \times N_2$ of keys as mentioned in Sect. 2.2, three schemes that manage less keys and subordinately derive $N_1 \times N_2$ of keys from the managed keys have been proposed [8, 9, 10].

The first scheme [8] controls access to JP2 codestreams according to the hierarchy in the prior scalability. This scheme, Scheme I hereafter, subordinately derives keys from the managed key using hash chains [13]. It, thus, requires five managed keys and five codestreams for five progression orders. The number of managed keys in Scheme I, $N_{\mathrm{m,I}}$, is

$$N_{\mathrm{m,I}} = 5. \tag{1}$$

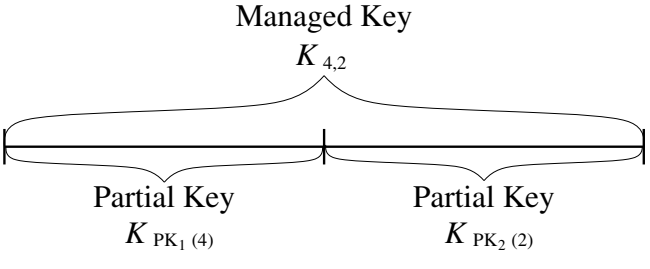


Fig. 5. Managed key consisting of two partial keys [7]

The second and third schemes [9], Scheme II and Scheme III hereafter, simultaneously control access in every hierarchical scalability with a single codestream. The number of managed keys in Scheme II, $N_{\mathrm{m,II}}$, is

$$N_{\mathrm{m,II}} = \min\left(N_1, N_2\right), \tag{2}$$

and the number of managed keys in Scheme III, $N_{\mathrm{m,III}}$, is

$$N_{\mathrm{m,III}} = N_1 + N_2 - 1, \tag{3}$$

whereas the proposed scheme needs only a single managed key.

3 Proposed Scheme

This section proposes a new scheme for access control to JP2 coded images that reduces the number of managed keys to one. The proposed scheme simultaneously controls access in every hierarchical scalability with a single managed key and a single managed codestream. The proposed scheme is resistant to collusion attacks as Schemes I, II, and III.

3.1 Key Derivation and Encipherment of Codestream

As an example of JP2 codestreams for explanation, the proposed scheme assumes the JP2 codestream shown in Fig. 2, where it is composed of five layers ($N_1 = 5$) and three resolution levels ($N_2 = 3$). The proposed scheme controls access regardless of progression orders.

Fig. 6 shows a new key derivation order, where K_{n_1,n_2} is the key for packet P_{n_1,n_2}. This order is resilient to collusion attacks. It is noted that key K_m is the single managed key.

Firstly in the proposed scheme, key $K_{4,2}$ is derived from K_m as

$$K_{4,2} = h\left(s\left(K_m\right)\right), \tag{4}$$

where $s(\cdot)$ is a cyclic shift and $h(\cdot)$ is a cryptographic one-way hash function. Replacing the combination of $s(\cdot)$ and $h(\cdot)$ with $f(\cdot)$, Eq. (4) is represented as

$$K_{4,2} = f\left(K_m\right). \tag{5}$$

Similarly, keys $K_{4,1}$ and $K_{4,0}$ are derived by

$$K_{4,n_2} = f^{3-n_2}\left(K_m\right), \qquad n_2 = 1, 0, \tag{6}$$

respectively, where $f^{\alpha}\left(\beta\right)$ represents that $f(\cdot)$ is applied to β recursively α times. The combinations of a cyclic shift and a hash function $f(\cdot)$ are shown with dashed arrows in Fig. 6.

Meanwhile, keys $K_{n_1,2}$ ($n_1 = 3, 2, 1, 0$) are derived by a hash chain. In this example, these keys are given as

$$K_{n_1,2} = h^{4-n_1}\left(K_{4,2}\right), \qquad n_1 = 3, 2, 1, 0, \tag{7}$$

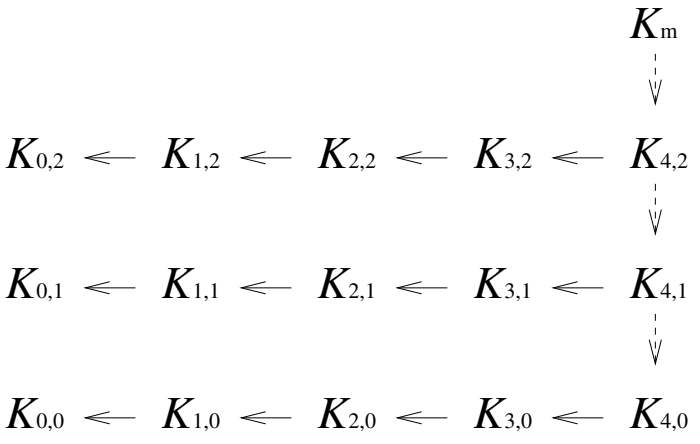


Fig. 6. Key derivation to control access to the JP2 codestream having five layers and three resolution levels ($N_1 = 5$ and $N_2 = 3$). K_{n_1,n_2} is the key for JP2 packet P_{n_1,n_2}. A solid arrow is an hash function and a dashed arrow is a combination of a cyclic shift and a hash function.

respectively, where $h^{\alpha}(\beta)$ represents that cryptographic one-way hash function $h(\cdot)$ is applied to β recursively α times. Similarly, keys $K_{n_1,1}$ and $K_{n_1,0}$ are derived by

$$K_{n_1,n_2} = h^{4-n_1}(K_{4,n_2}),$$
$$n_1 = 3, 2, 1, 0, \quad n_2 = 1, 0, \tag{8}$$

respectively. The hash chains are shown with solid arrows in Fig. 6.

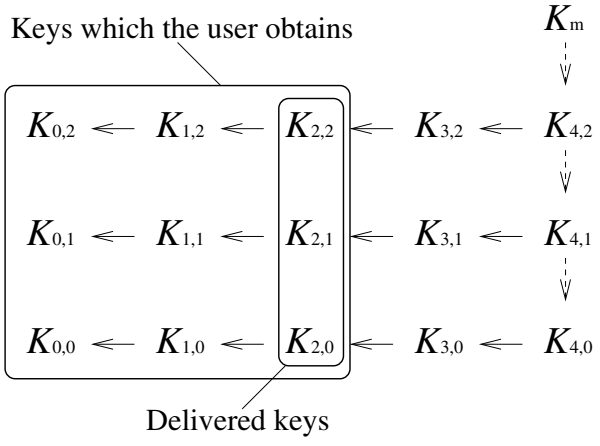
By introducing a combination of a cyclic shift and a hash function shown in Eq. (4), all keys K_{n_1,n_2} for JP2 packets P_{n_1,n_2} are derived from single managed key K_m.

With key K_{n_1,n_2}, JP2 packet P_{n_1,n_2} in the JP2 codestream is enciphered, where $n_1 = 0, 1, \ldots, N_1 - 1$ and $n_2 = 0, 1, \ldots, N_2 - 1$. It is noted that any arbitrary symmetric encipher algorithm can be used in the proposed scheme.

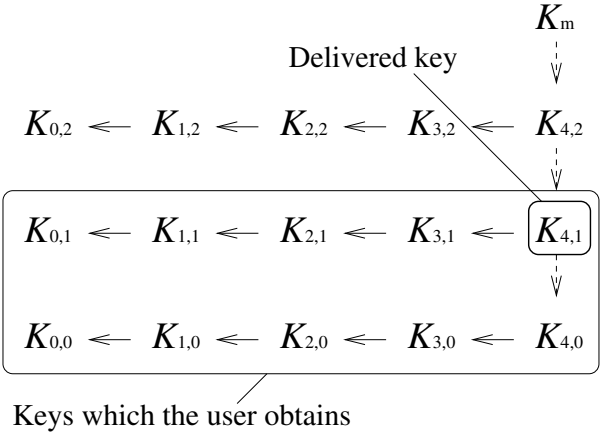
3.2 Decipherment and Decompression of Codestream

Here, it is considered that a user is allowed to access the image with quality $Q_{2,2}$, c.f. Fig. 3. The user receives keys $K_{2,2}$, $K_{2,1}$, and $K_{2,0}$ as shown in Fig. 7(a). To decompress the image at $Q_{2,2}$, the user needs to decipher nine packets $P_{0,0}$, $P_{0,1}$, $P_{0,2}$, $P_{1,0}$, $P_{1,1}$, $P_{1,2}$, $P_{2,0}$, $P_{2,1}$, and $P_{2,2}$. The six keys $K_{0,0}$, $K_{0,1}$, $K_{0,2}$, $K_{1,0}$, $K_{1,1}$, and $K_{1,2}$ that the user needs are derived from the delivered keys $K_{2,2}$, $K_{2,1}$, and $K_{2,0}$ as

$$K_{n_1,n_2} = h^{2-n_1}(K_{2,n_2}),$$
$$n_1 = 1, 0, \qquad n_2 = 2, 1, 0. \tag{9}$$



(a) Keys for $Q_{2,2}$.



(b) Keys for $Q_{4,1}$.

Fig. 7. Delivered and derived keys that the user needs to decompress the image at certain quality

By using keys $K_{0,0}$, $K_{0,1}$, $K_{0,2}$, $K_{1,0}$, $K_{1,1}$, $K_{1,2}$, $K_{2,0}$, $K_{2,1}$, and $K_{2,2}$, corresponding packets are deciphered and decompressed to present the image at $Q_{2,2}$.

As another example, it is assumed that a user can access the image with quality $Q_{4,1}$. The user receives single key $K_{4,1}$ as shown in Fig. 7(b). To access the image at $Q_{4,1}$, the user has to obtain ten keys $K_{0,0}$, $K_{0,1}$, $K_{1,0}$, $K_{1,1}$, $K_{2,0}$, $K_{2,1}$, $K_{3,0}$, $K_{3,1}$, $K_{4,0}$, and $K_{4,1}$. First, $K_{4,0}$ is derived from the delivered key $K_{4,1}$ as

Table 1. Comparisons in terms of the number of managed keys and delivery of managed keys

	Proposed	Scheme I [8]	Scheme II [9]	Scheme III [10]
The number of managed keys	1	5	$\min(N_1, N_2)$	$N_1 + N_2 - 1$
Delivery of managed keys	No	Yes	Yes	Yes

$$\begin{aligned} K_{4,0} &= h\left(s\left(K_{4,1}\right)\right) \\ &= f\left(K_{4,1}\right), \end{aligned} \tag{10}$$

which is the combination of a cyclic shift and a hash function. Then, eight keys K_{n_1,n_2} $(n_1 = 0, 1, 2, 3, \quad n_2 = 1, 0)$ are derived by

$$\begin{aligned} K_{n_1,n_2} &= h^{4-n_1}\left(K_{4,n_2}\right), \\ n_1 &= 3, 2, 1, 0, \qquad n_2 = 1, 0, \end{aligned} \tag{11}$$

and the user can obtain the ten keys for the ten packet.

3.3 Features

This section verifies that the proposed scheme meets requirements described in Sect. 2.3. The proposed scheme is evaluated by comparing with the conventional schemes [7, 8, 9, 10] which use hash chains [13] only.

Collusion Attack-Resistance. The proposed scheme is resilient to collusion attacks as well as the conventional schemes [8, 9, 10], i.e., Schemes I, II, and III, while the conventional scheme [7] is naive for collusion attacks.

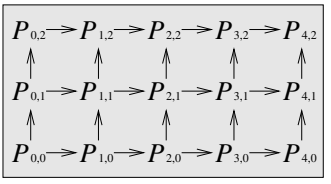
Alice and Bob appeared in Sect. 2.3 reappear here. Since Alice can access the image at $Q_{0,2}$, she receives keys $K_{0,2}$, $K_{0,1}$, and $K_{0,0}$. Bob receives single key $K_{4,0}$ to access the image at $Q_{4,0}$. Bob derives $K_{3,0}$, $K_{2,0}$, $K_{1,0}$, and $K_{0,0}$ from his delivered key $K_{4,0}$ by using Eq. (8). They obtain seven valid keys $K_{0,0}$, $K_{0,1}$, $K_{0,2}$, $K_{1,0}$, $K_{2,0}$, $K_{3,0}$, and $K_{4,0}$, but they can not derives any keys which they are not permitted to derive from these seven keys.

Thus, the proposed scheme is enough resistant to collusion attacks, though this paper does not explicate all pattern of collusion attacks.

Managed keys. Table 1 shows the results of comparisons in terms of the number of managed keys and delivery of managed keys. The proposed scheme manages only a single key regardless of the kinds of scalability and the depth of the hierarchy in each scalability, whilst Scheme I [8] must manage five keys and Scheme II [9] must manage keys as many as the minimum depth of hierarchy of two scalabilities. The number of managed keys in Scheme III [10] is just about the sum of the depth of two hierarchical scalabilities.

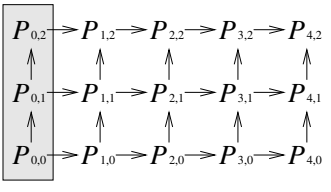
The single managed key is not delivered to any user in the proposed scheme, whereas the managed keys are delivered to some users in Schemes I, II, and III.

(a) Fully decompressed ($Q_{4,2}$). PSNR: 36.68 dB.



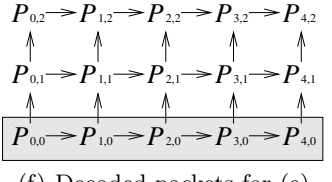
(b) Decoded packets for (a).

(c) Alice's ($Q_{0,2}$). PSNR: 27.71 dB.



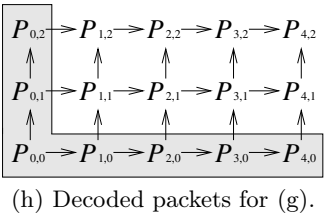
(d) Decoded packets for (c).

(e) Bob's ($Q_{4,0}$). PSNR: 29.51 dB.[1]



(f) Decoded packets for (e).

(g) Colluded. PSNR: 30.18 dB.



(h) Decoded packets for (g).

Fig. 8. Image examples. 512 × 512-sized lena is compressed. Five 0.1 bits/pixel-rate layers ($N_1 = 5$) and three resolution levels ($N_2 = 3$).

[1] Decompression of the LL subband and other subbands filled with zero.

4 Experimental Results

Grayscale image "lena" is compressed by Kakadu to generate a codestream with five layers ($N_1 = 5$) and three resolution levels ($N_2 = 3$). The bitrate of a layer is 0.1 bits/pixel, and Fig. 8 (a) shows the fully decompressed image, i.e., at quality $Q_{4,2}$. Alice can access the image with quality $Q_{0,2}$ shown in Fig. 8 (c), and Bob obtains the image shown in Fig. 8 (e) as $Q_{4,0}$. In the proposed scheme, Alice and Bob illegally derive the image shown in Fig. 8 (g). Since no illegally deciphered packet contributes the quality of this image, two users do not benefit from the collusion attack. Simulations with other images give similar results.

5 Conclusion

This paper has proposed a new key derivation scheme for access control to JP2 coded images in which combinations of a cyclic shift and a hash function are employed. The proposed scheme manages a single key and the single managed key is not delivered to any user. The proposed scheme also prevents malicious users to collude for accessing an images at higher quality much than that allowed by their permission.

References

1. Xie, D., Kuo, C.C.J.: Multimedia data encryption via random rotation in partitioned bit streams. In: Proc. IEEE ISCAS, pp. 5533–5536 (2005)
2. Zhang, Z., Sun, Q., Wong, W.C., Apostolopoulos, J., Wee, S.: Rate-distortion-authentication optimized streaming of authenticated video. IEEE Trans. Circuits Syst. for Video Technol. 17, 544–557 (2007)
3. Grosbois, R., Gerbelot, P., Ebrahimi, T.: Authentication and access control in the JPEG 2000 compressed domain. In: Proc. SPIE, vol. 4472, pp. 95–104 (2001)
4. Haggag, A., Ghoneim, M., Lu, J., Yahagi, T.: Progressive encryption and controlled access scheme for JPEG 2000 encoded images. In: Proc. IEEE ISPACS, pp. 895–898 (2006)
5. Shahid, Z., Chaumont, M., Puech, W.: Selective and scalable encryption of enhancement layers for dyadic scalable H.264/AVC by scrambling of scan patterns. In: Proc. IEEE ICIP, pp. 1273–1276 (2009)
6. Won, Y.G., Bae, T.M., Ro, Y.M.: Scalable Protection and Access Control in Full Scalable Video Coding. In: Shi, Y.Q., Jeon, B. (eds.) IWDW 2006. LNCS, vol. 4283, pp. 407–421. Springer, Heidelberg (2006)
7. Joye, M., Yen, S.M.: One-Way Cross-Trees and Their Applications. In: Naccache, D., Paillier, P. (eds.) PKC 2002. LNCS, vol. 2274, pp. 346–356. Springer, Heidelberg (2002)
8. Wu, Y., Ma, D., Deng, R.H.: Progressive protection of JPEG 2000 codestreams. In: Proc. IEEE ICIP, pp. 3447–3450 (2004)
9. Imaizumi, S., Fujiyoshi, M., Kiya, H.: Efficient collusion attack-free access control for multidimensionally hierarchical scalability content. In: Proc. IEEE ISCAS, pp. 505–508 (2009)

10. Imaizumi, S., Fujiyoshi, M., Abe, Y., Kiya, H.: Collusion attack-resilient hierarchical encryption of JPEG 2000 codestreams with scalable access control. In: Proc. IEEE ICIP, pp. II-137–II-140 (2007)
11. Information technology — JPEG 2000 image coding system – Part 1: Core coding system. ISO/IEC IS–15444–1 (2004)
12. Information technology — JPEG 2000 image coding system – Part 8: Secure JPEG 2000. ISO/IEC IS–15444–8 (2007)
13. Lamport, L.: Password authentication with insecure communication. Communications of the ACM 24(11), 770–772 (1981)