This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE CCNC 2006 proceedings.

An Efficient Key Scheme for Multiple Access of JPEG 2000 and Motion JPEG 2000 Enabling Truncations

Bin B. Zhu[1], Yang Yang[2], Shipeng Li[1]
[1]*Microsoft Research Asia, Beijing 100080, China*
[2]*Dept. of Elec. Eng. & Info Sci., Univ. of Sci. & Technol. of China, Hefei, Anhui 230027, China*
[1]*{binzhu, spli}@microsoft.com*

Abstract

JPEG 2000 provides multiple scalable accesses to a single codestream. Digital Rights Management of a JPEG 2000 codestream should preserve the original flexibility of scalability yet provide a mechanism to ensure what you see is what you pay: a low resolution version displayed on a smart phone should pay less than a high resolution version displayed on a PC. We present an efficient key scheme for multi-type, multi-level scalable access control for JPEG 2000 and motion JPEG 2000 codestreams. The scheme is based on a poset representation of the scalable access control and a hash based hierarchical access key scheme, both proposed elsewhere. The proposed key scheme exploits the information contained in a codestream and the features invariant under truncations to minimize the file size overhead for DRM applications yet preserve correct derivation of keys for descendants even when an encrypted codestream is truncated.

1 Introduction

JPEG 2000 (J2K) [1] is a newest image coding standard which provides high compression efficiency, lossy to lossless coding, and flexible scalability. A J2K codestream is organized in a hierarchical structure with packets as the fundamental building blocks. A J2K codestream supports Fine Granularity Scalability (FGS). Five scalable types, tile, resolution, quality, color-component, and precinct, are supported with a single codestream. A J2K codestream can be truncated to a preset layer (i.e. quality), resolution, component, set of tiles, and/or set of precincts. Motion JPEG 2000 which encodes each video frame independently is also defined in the standard [2].

Protection of multimedia content from unauthorized access has been actively studied in the past decade. JPEG 2000 has recently published the security part of the standard, JPSEC [3]. Digital Rights Management (DRM) is usually employed to provide persistent protection throughout the life of multimedia content from creation to consumption. A DRM system is usually based on encryption. A JPEG 2000 codestream consists of many fundamental building blocks called packets. A packet is uniquely specified by tile, resolution level, component, precinct, and layer. Different keys should be used to encrypt different packets to support multi-type, multi-level scalable access control, which is very different from encryption of non-scalable codestreams where a single key is used for the whole codestream. A multi-type, multi-level scalable access control system ensures fair consumption of scalable content: what you see is what you pay. A portable device with a small display screen pays less to show an image than a PC with a large, high resolution display screen. Therefore key management for scalable access control of a scalable codestream is much more complex than that of a non-scalable codestream. Additional complexity comes from the fact that for some access types, the access to a level is entitled to access all the lower levels of the same type. For example, the access to a resolution level in J2K should also be able to access all the resolution levels below it. A key scheme should support such a hierarchy to minimize the number of keys needed to deliver to a client.

Several key schemes for scalable access control have been reported. The simplest scheme is the one used in [4] for MPEG-4 FGS codestreams where each independent encrypted block in a frame is associated with an independent key. All the keys associated with the encryption blocks that a specific access is entitled to access have to be sent to a client. A tree-based key scheme is proposed in [5] for scalable access control of J2K codestreams. A single key may be sent to a user for certain accesses. An efficient key scheme based on the Diffie-Hellman (DH) key agreement is proposed in [6] for the two access types described in [4], where a single key is sent to a user. The scheme was then generalized to support an arbitrary number of access

1-4244-0086-4/05/$20.00 ©2005 IEEE.

types [7]. A recent paper [8] classifies scalable access types into two categories: fully ordered hierarchic type and partially ordered hierarchic type. For a J2K codestream, resolutions, layers, and component are of the fully ordered hierarchic type, while tiles and precincts are of the partially ordered hierarchic type. A fully ordered hierarchic type is represented by a fully ordered set (foset) and a partially ordered hierarchic type is represented by a partially ordered set (poset). The product of the scalable access types is a poset which is the desired access control of the scalable codestream. Many key schemes developed for hierarchical access control of a poset can be then used as a key scheme for the multi-type, multi-level scalable access control DRM system.

One of the efficient key schemes for hierarchical access control of a poset is the hash-based key scheme proposed in [9][10] where each node in a poset is associated with a pair of numbers, a key and a unique label. Each edge is also assigned a number. Node labels and edge numbers are public information and are used to derive the keys of descendants of a node with known key. In a DRM system, public information is packed into the protected content which can be easily accessed by a user to derive the keys needed for decryption. The key of the highest node a user has access right to is usually packed into a license together with the rights specification and derived to a user.

With the poset generated with the scheme described in [8] and the hash-based hierarchical access key scheme described in [9][10], a single key is delivered in a license to a user. An important issue to be solved is then to minimize the public information packed in an encrypted J2K or motion J2K codestream, i.e., to minimize the file size overhead due to the key scheme for multi-type, multi-level scalable access control. In addition, the key scheme should enable truncation of an encrypted J2K or motion J2K codestream directly without decryption. In other words, keys can still be correctly derived for a truncated encrypted codestream. The two requirements are typically conflicting with each other: correct derivation of keys from the public information usually requires insertion of additional information so that the original decryption keys can still be correctly derived even when an encrypted codestream is truncated, resulting in larger overhead.

In this paper, we propose an efficient key scheme for multi-type, multi-level scalable access control for JPEG 2000 and motion JPEG 2000 codestreams. The scheme is based on the poset generated with the scheme in [8] and the hash based hierarchical access key scheme in [9][10]. To maintain correction derivation of keys of all the descendants for each node in the poset while minimizing overhead due to public information packed in an encrypted codestream, invariant features of scalable access types are employed in the key generation scheme. More specifically, data derived from canvas coordinates in JPEG 2000 is used by the hash based hierarchical key scheme to generate node keys which are then used to generate encryption keys. Canvas coordinates can be derived from the information contained in a J2K codestream, and therefore does not add any overhead. On the other hand, canvas coordinates are invariant to the aforementioned truncations. Our scheme to be proposed in this paper has solved the conflicting requirements of minimizing overhead and correct key derivation after truncation of an encrypted codestream.

This paper is organized as follows. In Section 2 we present the background needed for describing our proposed key scheme. JPEG 2000 and motion JPEG 2000 are briefly described in this section, followed by the scheme to generate a poset to represent a multi-type, multi-level scalable access control. The section concludes with a brief description of a hash-based key scheme for hierarchical access control. In Section 3 our proposed key scheme is described in detail. The paper concludes with Section 4.

2 Background

2.1 JPEG 2000 and Motion JPEG 2000

In J2K, an image can be partitioned into rectangular regions called tiles. Each tile is encoded independently. Data in a tile are divided into one or more components in a color space. A wavelet transform is applied to each tile-component to decompose the image data into different resolution levels. The lowest frequency subband is referred to as the resolution level 0 subband, which is also resolution 0. The image at resolution r (r>0) consists of the data of the image at resolution (r-1) and the subbands at resolution level r. Bitstreams from code-blocks are distributed across one or more layers in the codestream. Each layer represents a quality increment. A layer consists of a number of consecutive bit-plane coding passes from each code-block in the tile, including all subbands of all components for that tile. J2K also provides an intermediate space-frequency structure known as the precinct. A precinct is a collection of spatially contiguous code-blocks from all subbands at a particular resolution level. The fundamental building block in a J2K codestream is called a packet, which is simply a continuous segment in the compressed codestream that consists of a number of bit-plane coding passes from each code-block in a precinct. Each packet is uniquely identified by the five scalable

parameters: tile, component, resolution level, layer, and precinct.

Canvas coordinate system is used in JPEG 2000. An image is bounded by its upper left hand corner of coordinates (a_x, a_y), and its bottom right hand corner of coordinates (b_x, b_y). Let (P_x, P_y) be the coordinates of the top left corner of the first tile on the high resolution grid, then J2K requires that $P_x, P_y \geq 0$; $P_x \leq a_x$, $P_y \leq a_y$; and $P_x + T_x > a_x$, $P_y + T_y > a_y$, where $T_y \times T_y$ is the size of a tile at the high resolution grid. Once the coordinates of the first tile is known, the coordinates of the rest tiles can be derived. Coordinates for precincts are similarly defined, and those of lower resolution grid can be derived. For details of JPEG 2000 and motion JPEG 2000, readers are referred to [1][2].

2.2 POSET for Multi-type, Multi-level Scalable Access Control

Multi-type, multi-level scalable access control for JPEG 2000 and motion JPEG 2000 can be represented by a poset. To generate such a poset, levels of each scalable access type is ordered, either fully or partially. Figure 1(A) and Figure 1(B) show an example of the Hasse diagrams for the two types of sets: one is two resolutions r_1 and r_2, and the other is three tiles t_0 , t_1 , and t_2 . The resulting poset representing the access control on the resolutions and tiles is shown in Figure 1(C). The details of generating a poset for multi-type, multi-level access control of JPEG 2000 and motion JPEG 2000 can be found in [8].

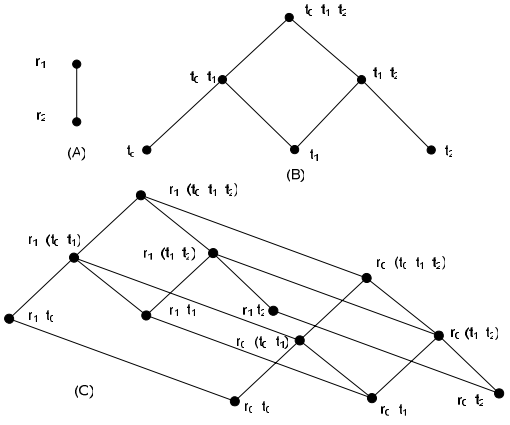

Figure 1: Hasse diagrams for fully ordered set of 2 resolutions (A), partially ordered set of tiles t_1, t_2, t_3 (B), and the resulting poset representing the access control on resolutions and tiles (C).

2.3 Hash-Based Key Scheme for Hierarchical Access Control

Many key schemes have been proposed for hierarchical access control of a poset. One of the most efficient key schemes is the hash-based key scheme proposed in [9][10]. In this scheme, each node n_i is assigned a unique label l_i and a secret key k_i . Suppose there is an edge e_{12} linking a node n_1 to another node n_2 , where $n_1 > n_2$, then the edge is assigned a value $v_{12} = k_2 - H(k_1, l_2)$, where H is a secure cryptographic hash function. The labels l_i and edge values $v_{i,j}$ are public information which can be packed with the encrypted codestream. Figure 2 shows an example of a poset with 4 nodes. To derive the key of a child node, the label of the child node and the edge linking from the current node to the child node is used. For example, if the key k_1 of the node n_1 in Figure 2 is known. The key k_3 for a child node n_3 can be derived with the following equation:

$$k_3 = v_{1,3} + H(k_1, l_3),$$

from key k_1 of node n_1 and public edge value $v_{1,3}$ and label l_3 . Both $v_{1,3}$ and l_3 are public information. Details for the hash-based key scheme can be founds in [9][10].

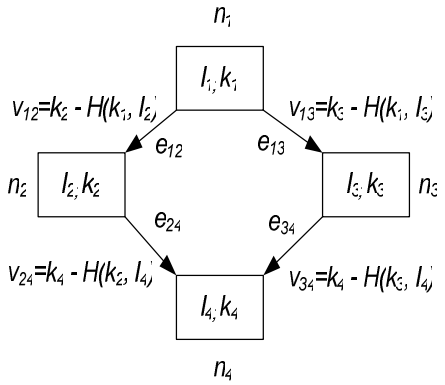

Figure 2: Node and edge values for a poset. All arithmetic is module ρ which is a proper number.

3 Efficient Key Scheme Enabling Direct Truncations of Encrypted Codestreams

Since the number of keys delivered to a user in a license is always 1 for the scheme, our scheme to be described in detail here focuses on minimizing the overhead caused by the public information packed with

an encrypted JPEG 2000 or motion JPEG 2000 codestream, while maintaining correction derivation of keys for descendants after truncation of the encrypted codestream. In a J2K codestream, tiles are indexed with a unique integer starting from the upper left tile. If some tiles are truncated, say the surrounding tiles are truncated to convert an image or frame of aspect 16:9 to 4:3, the tiles in the resulting J2K codestream will be indexed differently from the original indexes. Therefore the tile indexes used in J2K codestreams cannot be used as the unique labels l_i in key generation, otherwise the keys generated from truncated codestreams will be different from the actual decryption keys. In our scheme, canvas coordinates are used in generating unique labels l_i which are invariant when truncations occur. Therefore the values of labels and edges are preserved even if an encrypted codestream undergoes truncations. The poset structure can be derived from the information carried in a J2K codestream. Only the edge values are required to be packed into an encrypted codestream in our proposed scheme, dramatically reducing the overhead due to auxiliary public information.

To exploit the invariance of canvas coordinates yet satisfy the requirements set by JPEG 2000 on tiles, as described in Section 2.1, a truncation-invariant virtual coordinate system to uniquely identify each tile is proposed. To generate this virtual coordinate system, an image and its tiles are extended upwards and leftwards in the following way until it cannot move any further: each time the coordinates (a'_x, a'_y) of top left corner of the extended image and the coordinates (P'_x, P'_y) of the extended first tile are moved upwards and leftwards stepwise. Each move reduces the coordinates of both points (a'_x, a'_y) and (P'_x, P'_y) in either x or y direction by subtracting the corresponding T_x or T_y, where $T_y \times T_y$ is the size of a tile at the high resolution grid. For the resulting virtual coordinate system, we have $P'_x, P'_y \geq 0$, $a'_x, a'_y \geq 0$, $P'_x < T_x, P'_y < T_y$, $P'_x \leq a'_x, P'_y \leq a'_y$, and $P'_x + T'_x > a'_x, P'_y + T'_y > a'_y$. Figure 3 shows an example of the actual image, its tiles (solid lines), the extended image, and extended tiles (dotted lines indicating virtual tiles and virtual image areas).

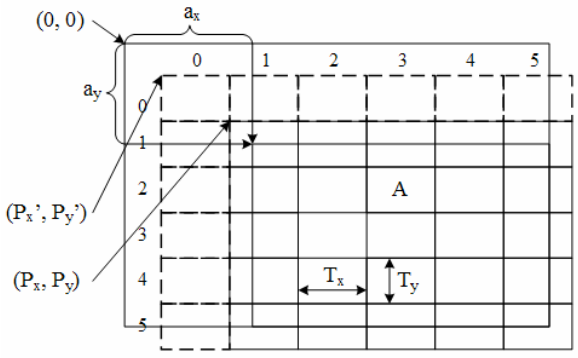

Figure 3: A virtual coordinate system for extended image and tiles.

After the above extension and generation of the virtual coordinate system, tiles are indexed, with both virtual and real tiles counted, starting from 0 on both x- and y- directions, from top to bottom and left to right. We use two binary arrays to index a tile in an image. Each array is associated with a direction, and the *i*-th bit in the binary array is associated with the tile of index *i* in the corresponding direction. For example, the tile "A" shown in Fig. 3 is indicated by the two binary arrays 00010…, and 00100…. It is clear to see that in this tile indexing system, a combination of tiles can be simply represented by setting the corresponding bits in the two binary arrays. Therefore the component associated with the tile access for each node of the poset representing a multi-type, multi-level scalable access control of JPEG 2000 or motion JPEG 2000 can be represented by the two binary arrays. We note that such a tile indexing system does not change even if a JPEG 2000 codestream undergoes truncations.

Having designed an invariant indexing system for tiles, let us turn attention to other scalable types. It is easy to see that the indexes used in JPEG 2000 to represent resolution levels, layers, and components do not change after truncations. The index for precincts does not change either, even if some precincts can be truncated. This is because indexes of precincts are generated against the coordinates of the corresponding bands, which in turn against the coordinates of the corresponding tiles, and therefore invariant after truncations.

The aforementioned truncation-invariant indexing system is used to indicate each node in a poset representing a multi-type, multi-level scalable access control of JPEG 2000 or motion JPEG 2000. The index for each node is unique, and therefore can be used as the labels $\{l_i\}$ in the hash-based key scheme described in Section 2.3. We note that these indexes can be generated from the information contained in the headers of a J2K or motion J2K codestream, and

therefore are not needed to be packed into a codestream. The poset corresponding to a multi-type, multi-level scalable access control can also be derived from the headers in a codestream, and is not packed into a codestream. The only public information our scheme has to pack into a codestream is the edge values of the poset. This is much less than the information packed into a codestream using other schemes such as the scheme proposed in [8], resulting in less overhead.

For motion JPEG 2000, the key scheme for an image is used for a frame. The keys are reused for each frame unless they are rekeyed.

Figure 4 shows an image of aspect ratio 16:9 cropped to 4:3 by truncating the surrounding tiles. This type of truncations is widely applied to show a movie content of aspect ratio 16:9 on a TV of aspect ratio 4:3.

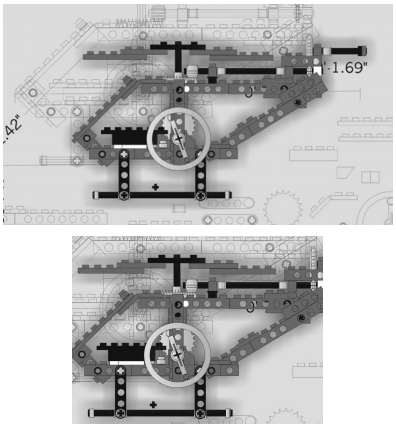

Figure 4: Cropping from aspect ratio 16:9 (1280 by 720 pixels) to 4:3 (792 by 594 pixels).

4 Conclusion

We have proposed an efficient key scheme for multi-type, multi-level scalable access control for JPEG 2000 and motion JPEG 2000 codestreams. The scheme is based on the poset generated with the scheme in [8] and the hash based hierarchical access key scheme in [9][10]. The proposed key scheme exploits the information contained in a codestream and the features invariant under truncations to minimize the file size overhead for DRM applications yet preserve correction derivation of keys for descendants even when an encrypted codestream is truncated.

References

[1] *Information Technology – JPEG 2000 Image Coding System, Part 1: Core Coding System*, ISO/IEC 15444-1:2000 (ISO/IEC JTC/SC 29/WG 1 N1646R, March 2000).

[2] *Information Technology – JPEG 2000 Image Coding System, Part 3: Motion JPEG 2000*, ISO/IEC 15444-3:2002.

[3] F. Dufaux, S. Wee, J. Apostolopoulos and T. Ebrahimi, "JPSEC for Secure Imaging in JPEG 2000," *SPIE Proc. Applications of Digital Image Processing XXVII*, vol. 5558, pp. 319-330, Nov. 2004.

[4] B. B. Zhu, C. Yuan, Y. Wang, and S. Li, "Scalable Protection for MPEG-4 Fine Granularity Scalability," *IEEE Trans. on Multimedia*, vol. 7, no. 2, pp. 222-233, April 2005.

[5] R. H. Deng, Y. Wu, and D. Ma, "Securing JPEG2000 Code-Streams," *Int. Workshop Adv. Dev. in Software & Systems Security*, Dec. 2003.

[6] B. B. Zhu, M. Feng, and S. Li, "An Efficient Key Scheme for Layered Access Control of MPEG-4 FGS Video," *IEEE Int. Conf. Multimedia & Expo,* vol. 1, pp. 443-446, Taiwan, June 2004.

[7] B. B. Zhu, M. Feng, and S. Li, "A Framework of Scalable Layered Access Control for Multimedia," *IEEE Int. Symp. Circuits and Systems 2005*, May 2005, pp. 2703-2706.

[8] B. B. Zhu, M. Feng, and S. Li, "Secure Key Management for Flexible Digital Rights Management of Scalable Codestreams," to appear in *IEEE Int. Workshop on Multimedia Signal Processing (MMSP) 2005.*

[9] S. Zhong, "A Practical Key Management Scheme for Access Control in a User Hierarchy," *Computer & Security*, vol. 21, no. 8, pp. 750-759, 2002.

[10] K. Frikken, M. Atallah, and M. Bykova, "Hash-Based Access Control in an Arbitrary Hierarchy," *CERIAS Technical Report 2004-49, Purdue University*, Nov. 2004.