Codestream Level Secure Identification for JPEG 2000 Images under Various Compression Ratios

Kenta Iida* and Hitoshi Kiya*
* Tokyo Metropolitan University, Hino, Tokyo, 191-0065, Japan
Email:iida-kenta1@ed.tmu.ac.jp, kiya@tmu.ac.jp
phone number:81-42-585-8454

***Abstract*—A secure identification scheme for JPEG 2000 codestreams is proposed in this paper. The aim is to securely identify JPEG 2000 images generated from the same original image, without decoding images. Features used for the identification are extracted from header parts in a JPEG 2000 codestream. The proposed scheme does not provide any false negative matches under various compression ratios, while most of image hashing-based schemes do not guarantee this performance. Existing identification schemes that do not provide any false negative matches can not be securely carried out. Due to such a situation, we propose an identification system based on a fuzzy commitment scheme, which is a well-known secure protocol for biometric template protection. Moreover, an error correction technique with 1-bit parity is considered to achieve the system. The experiment results show the proposed scheme is effective in terms of true positive matches, while keeping the security high.**

I. Introduction

The use of images and video sequences has greatly increased recently because of the rapid growth of the Internet and multimedia systems. It is often necessary to identify a certain image in a database that has a large number of digital images in various types of applications. The image database generally consists of images in a compressed form to reduce the amount of data. In addition, most of the contents include sensitive information such as personal data and copyright [1], [2]. "Identification" in this work is defined as an operation for finding an image that is identical to a given original image from an image database. In this paper, a robust scheme against a difference of compression ratios for identifying JPEG 2000 images securely is proposed.

JPEG 2000 is an image coding system standardized by ISO/IEC. Besides, JPEG 2000 has been officially selected as the standard compression/decompression technology for digital cinema by the Digital Cinema Initiatives consortium [3]. For identifying a large number of frames, codestream-based identification methods for JPEG 2000 images have been proposed [4]–[7] for editing cinema films, which have sensitive data. The identification can be performed without decoding images. Some of conventional schemes do not provide any false negative matches, but secure identification is not considered.

Several identification schemes have been developed for not only JPEG 2000 but also JPEG and JPEG XR images [8]–[16]. However, most of these schemes do not consider to securely identify images. Moreover, a lot of image hash functions have been presented [17]–[19]. However, the hashing-based identification schemes do not guarantee that there are not false negative matches and have to decode compressed images to extract the features, although some schemes aim to securely identify images. Recently, secure image identification systems based on a fuzzy commitment scheme have been proposed [12], [16]. These systems do not provide any false negative matches. However, they are not codestream level identification schemes, so that decoding images is needed. Moreover, they are not available for JPEG 2000 images.

Because of such situations, a codestream level secure identification scheme is proposed. The identification can be performed not only at codestream level but also securely. To achieve the identification at codestream level, some features extracted from header parts in a JPEG 2000 codestream are used. Besides, for identifying images securely, the system based on the fuzzy commitment scheme [20] is considered. Moreover, the proposed scheme does not produce any false negative matches. The experimental results show the proposed scheme has a high query performance in terms of true positive matches while keeping the security high.

II. Preliminaries

A. Scenario of Image Identification

The image identification model considered in this paper is briefly illustrated in Fig.1. In the enrollment process, after analyzing a JPEG 2000 codestream, some features are extracted from the header parts in the codestream, without any decoding process. To be securely enrolled in a database, the features are encrypted. The encrypted features are sent to an authentication server and stored in the database. In the authentication process, the features are extracted from a query image after parsing the codestream as well as in the enrollment process, and then they are used to decrypt the data enrolled in the database. The authentication server identifies whether the query and each enrolled image are generated from the same original image or not by using the decrypted data. This is important for digital cinema systems because the identification system used for them must be able to handle the large number of frames encoded with the JPEG 2000. In the proposed system, a fuzzy commitment scheme [20] is applied to encrypt the data. As a result, any images and the raw features are not sent to the database.

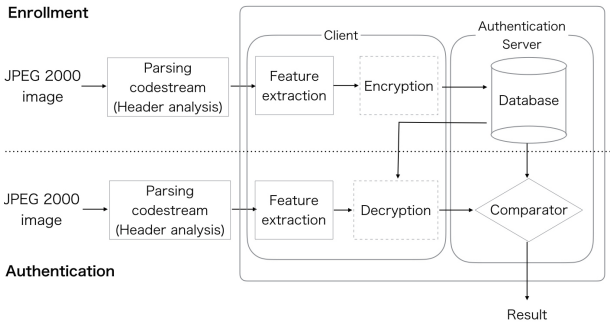


Fig. 1. Image identification model

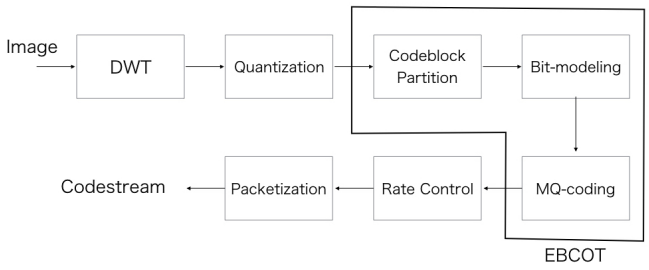


Fig. 2. Block diagram of JPEG 2000 encoder

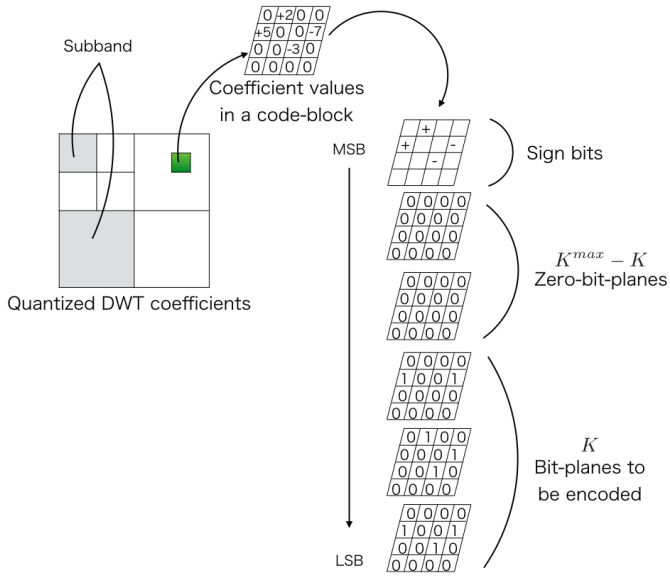


Fig. 3. Bit-plane decomposition and sign-magnitude. Representation of DWT coefficients in code-block. A zero-bit-plane is a special bit-plane in which the samples are all zeros. Zero-bit-planes are arranged from the MSB to the LSB level.

B. JPEG 2000 Coding and Zero Bit Planes

In this paper, the number of zero bit planes, which is extracted from a JPEG 2000 codestream, plays an important role for the image identification.

A block diagram of the JPEG 2000 encoder is shown in Fig.2. JPEG 2000 uses a bit-plane architecture summarized as follows. After DWT(Discrete Wavelet Transform) coefficients are quantized and grouped in code-blocks, they are represented in a sign-magnitude form. The sign bit-plane is at the MSB level, and the magnitude bit-planes are beneath it. The number of samples in a bit-plane is equal to size in the code-block, and all the samples in a bit-plane are either 0 or 1. The quantized magnitudes have K^{max}-bit representation. The block coder for JPEG 2000 first determines the number of bits K, $K \leq K^{max}$, that are needed to represent the quantized magnitudes. The difference $K^{max} - K$ is defined as follows.

$$K^{msbs} = K^{max} - K \tag{1}$$

K^{msbs} represents the number of the most significant magnitude bits that is skipped to encode with the encoder. The decoder will take this to be zero for all samples. This is called "number of zero-bit-planes (NZBP)". A code-block in which all bit-planes are zero-bit-planes in the JPEG 2000 standard is defined as "not included" because the code-block does not contain any data to be encoded.

The coding-rate in the JPEG 2000 is normally controlled by discarding the MQ-encoded codestream from LSB to MSB in rate control. According to coding-rate, the level discarding bit-planes is determined and then all bit-planes under it are treated as zero-bit-planes. Thus, there is fundamentally no effect from NZBP even if the coding rate changes. However, the number of "not included" code-blocks may change due to the bit plane truncation. Therefore, this property has been used in a number of robust identification schemes for JPEG 2000 images [4]–[7].

NZBP is in the header information of a JPEG 2000 codestream. It is easily obtained by parsing the header parts without decoding the image.

Several notations used in the following section are listed here.

- X represents an image.X can be "Q" for a query image and "O" for the original image, where all images have the same size.
- L_1 represents the number of code-blocks in an image.
- $X(k)$ represents NZBP of kth code-block in image X.$0 \leq k < L_1$.
- $X(k) = -1$ means that NZBP of kth code-block is "not included".

III. Proposed Scheme

A. Property of NZBP

In this paper, NZBP is extracted from all code-blocks in a JPEG 2000 image because NZBP can be directly extracted from a JPEG 2000 codestream and has the following property [4]–[7].

- When JPEG 2000 images Q and X_i are generated from the same original image O with the same quantization step size, $Q(k)$ is equal to $X_i(k)$, even if the compression ratios are different. Namely, the relation is given as

$$Q(k) = X_i(k), (0 \leq k < L_1) \tag{2}$$

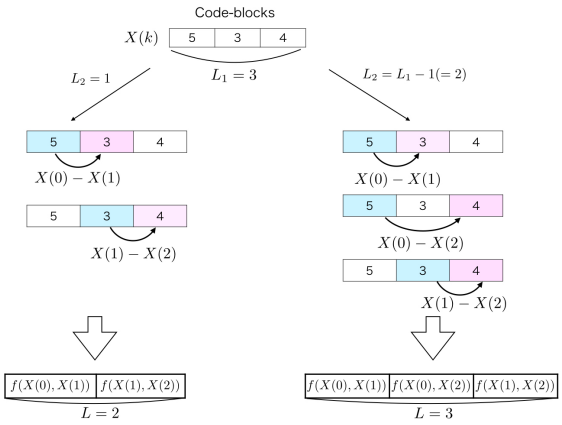


Fig. 4. Examples of feature generation

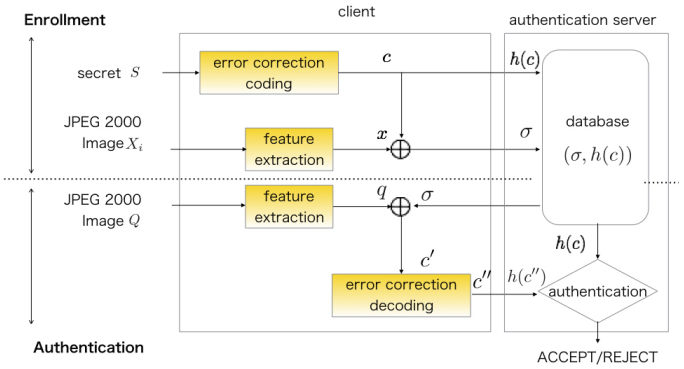


Fig. 5. Proposed identification system

where $Q(k)$ and $X_i(k)$ are not "not included".

When NZBP is directly mapped to a codeword to be encrypted, $K^{max}+1$ codewords are needed, so that each codeword requires the length corresponding to $K^{max}+1$. Now, to map NZBP to compact codewords, we extend Eq.(2) as below.

- When JPEG 2000 images Q and X_i are generated from the same original image O with the same quantization step size, the difference $Q(k) - Q(g)$ is also equal to $X_i(k) - X_i(g)$, where $0 \le g < L_1$. Namely, the relation is given as

$$Q(k) - Q(g) = X_i(k) - X_i(g), (0 \le k < L_1,\ 0 \le g < L_1) \tag{3}$$

where $Q(k)$, $Q(g)$, $X_i(k)$ and $X_i(g)$ are not "not included".

This property will be used to generate a compact feature. As described in II-B, the number of "not included" code-blocks depends on compression ratios. Note that it causes a problem that Eq. (3) can not be applied.

B. Compact Feature Generation

In the proposed scheme, an error correction technique is applied to avoid the effect of a difference in compression ratios. To apply the error correction method, the features need to be mapped to 3-bit codewords, so that a function is defined as below, where $k < g$.

$$f(X(k), X(g)) = \begin{cases} (000)_2, \text{if } X(k) - X(g) > 0, \\ \qquad X(k) \neq -1 \text{ and } X(g) \neq -1, \\ (101)_2, \text{if } X(k) - X(g) = 0, \\ \qquad X(k) \neq -1 \text{ and } X(g) \neq -1, \\ (110)_2, \text{if } X(k) - X(g) < 0, \\ \qquad X(k) \neq -1 \text{ and } X(g) \neq -1, \\ (001)_2, \text{if } X(k) = -1 \text{ or } X(g) = -1. \end{cases} \tag{4}$$

Note that the Hamming weight of $f(X(k), X(g))$, denoted by $W_H(f(X(k), X(g)))$ is an odd only when $X(k) = -1$ or $X(g) = -1$.

In the proposed system, 3-bit codewords in Eq. (4) are generated as shown in Fig. 4. L_2 represents the number of code-blocks compared with each code-block. $L_2 = 1$ means NZBP of a code-block is compared with only that of the next code-block. Namely, $f(X(k), X(k+1))$ is calculated for $L_2 = 1$. On the other hand, $L_2 = L_1 - 1$ means the NZBP of a code-block is compared with those of all other code-blocks as illustrated in Fig.4. Therefore, L represents the number of 3-bit codewords, that is determined by L_1 and L_2.

C. Proposed Identification System

Figure 5 shows the proposed identification system with a fuzzy commitment scheme. A new error correction technique is also proposed for this system.

1) Scenario of Proposed Scheme

Image X_i is sent to client to be enrolled in a database. After client extracts NZBP from X_i, to protect both X_i and NZBP, a commitment σ and a hash value $h(c)$ are calculated by using a secret code S and an error correcting code c respectively, where $h(\cdot)$ is a hash function. The set of σ and $h(c)$ is sent to an authentication server. Note that only the protected set is enrolled in the database. On the other hand, the image Q is sent to client to identify images generated from the same original image. Client extracts NZBP from Q and error-correction decoding is carried out to obtain c'' by using NZBP and σ. The hash value $h(c'')$ is then calculated and compared with $h(c)$ in the encrypted domain.

2) Enrollment Process

In order to securely enroll features of image X_i, the following steps are carried out.

(a) Set the values L_1 and L_2.
(b) Set $j := 0$, $k := 0$ and $g := 1$.
(c) Extract NZBP from all code-blocks in X_i.
(d) Generate a codeword $x(j)$ with 3 bits by using the set of NZBP $X_i(k)$ and $X_i(g)$ as

$$x(j) = f(X_i(k), X_i(g)) \tag{5}$$

(e) Select randomly a 2-bit secret code $s(j) \in S = \{(00)_2, (01)_2, (10)_2, (11)_2\}$ and map it to

a 3-bit error-correcting codeword $c(j) \in C = \{(000)_2, (011)_2, (101)_2, (110)_2\}$ respectively, by adding 1-bit parity to $s(j)$. For instance, $s(j) = (01)_2$ is mapped to $c(j) = (011)_2$. Note that the Hamming weight of $c(j)$, denoted by $W_H(c(j))$, is either two or zero.

(f) Calculate a commitment codeword $\sigma(j)$ as

$$\sigma(j) = x(j) \oplus c(j) \tag{6}$$

where $\oplus$ denotes the bit wise XOR operation.

(g) Set $g := g+1$ and $j := j+1$.If $g < L_1$ and $g < k+L_2+1$, proceed to step(d).

(h) Set $k := k+1$ and $g := k+1$.If $k < L_1 - 1$, proceed to step(d). Otherwise, generate codewords c and σ by connecting each component as below.

$$\sigma = \begin{cases} \sigma(0)\ldots||\sigma(j-1), \text{if } j \neq 1, \\ \sigma(0), \text{otherwise.} \end{cases} \tag{7}$$

$$c = \begin{cases} c(0)||\ldots||c(j-1), \text{if } j \neq 1, \\ c(0), \text{otherwise.} \end{cases} \tag{8}$$

(i) Calculate a hash value $h(c)$ for c and send $h(c)$ and σ to an authentication server. The set is stored in the server.

3) Authentication Process

In order to compare image Q with image X_i, the following steps are carried out.

(a) Set the values L_1 and L_2.

(b) Set $j := 0$, $k := 0$ and $g := 1$.

(c) Request the authentication server to send the commitment σ.

(d) Extract NZBP from all code-blocks in Q.

(e) Generate a codeword $q(j)$ with 3-bits by using the set of NZBP $Q(k)$ and $Q(g)$ as

$$q(j) = f(Q(k), Q(g)) \tag{9}$$

(f) Compute $c'(j)$ as

$$c'(j) = \sigma(j) \oplus q(j) \tag{10}$$

and apply error correction decoding to $c'(j)$ to obtain $c''(j)$(see III-D).

(g) Set $g := g+1$ and $j := j+1$.If $g < L_1$ and $g < k+L_2+1$, proceed to step(e).

(h) Set $k := k+1$ and $g := k+1$.If $k < L_1 - 1$, proceed to step(e). Otherwise, generate a codeword c'' by connecting each component as below:

$$c'' = \begin{cases} c''(0)||\ldots||c''(j-1), \text{if } j \neq 1, \\ c''(0), \text{otherwise,} \end{cases} \tag{11}$$

and compute a hash value $h(c'')$ and send it to the authentication server.

(i) Output "ACCEPT" if $h(c'') = h(c)$.

TABLE I
ERROR CORRECTION

	Error-correcting decoding	
$W_H(\sigma(j))$	$c''(j) =$	$c(j) = c''(j)$?
even	$c'(j)$	yes, if $x(j) = q(j)$
odd	$c'(j) \oplus (001)_2$, for $q(j)) = (000)_2$	yes
	$c'(j) \oplus (100)_2$, for $q(j) = (101)_2$	
	$c'(j) \oplus (111)_2$, for $q(j) = (110)_2$	
	$c'(j)$, for $q(j) = (001)_2$	

D. Error Correction Decoding

$c'(j)$ is mapped to $c''(j)$ for error correction as shown in step (e). Let us explain the error correction decoding in more detail. The proposed codeword $c(j)$ is designed to satisfy the condition:

$$c(j) \begin{cases} = c''(j), \text{if } x(j) = q(j) \\ \qquad\qquad \text{and } x(j) \neq (001)_2, \\ = c''(j), \text{if } x(j) = (001)_2, \\ \neq c''(j), \text{otherwise.} \end{cases} \tag{12}$$

1)$W_H(\sigma(j))$ is an even

When $W_H(\sigma(j))$ is an even, $c'(j)$ is mapped to $c''(j) = c'(j)$ (see Table I). Since $W_H(c(j))$ is absolutely an even, $W_H(\sigma(j)) = W_H(x(j) \oplus c(j))$ must be an even for $x(j) \neq (001)_2$ from Eq.(5). Thus, $c'(j) = \sigma(j) \oplus q(j)$ is equal to $c(j)$ under $x(j) = q(j)$ as Eq.(12). $c'(j) = c(j)$ is not guaranteed under $x(j) \neq q(j)$.

2)$W_H(\sigma(m,n))$ is an odd

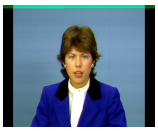
Since $W_H(\sigma(j))$ is an odd for only $x(j) = (001)_2$ from Eq.(5), $c'(j)$ is able to be mapped to $c''(j) = c(j)$ by the operations in Table I. For example, $c''(j)$ is mapped to $c'(j) \oplus (001)_2$ for $q(j) = (000)_2$ because of $c'(j) = x(j) \oplus c(j) \oplus q(j) = (001)_2 \oplus c(j) \oplus (000)_2 = (001)_2 \oplus c(j)$.

Similarly, $c'(j)$ is mapped to $c''(j) = c'(j) \oplus (101)_2$ for $q(j) = (100)_2$ and $c'(j)$ is mapped to $c''(j) = c'(j) \oplus (111)_2$ for $q(j) = (110)_2$. Moreover, $c'(j)$ is mapped to $c''(j) = c'(j)$ for $(001)_2$.

The above mapping operations do not guarantee to provide the correct relation $c''(j) = c(j)$ in any other cases such as $x(j) \neq q(j)$.

IV. EVALUATION FOR SECURITY

Several attacks have been introduced in [21] for the fuzzy commitment scheme, and studies on security analysis have been also considered in [20], [21]. In this section, to evaluate the safety of proposed scheme, we assume that attackers intend to deliver the brute force attack on $h(c)$. The aim of the attack is to estimate NZBP from information enrolled in the database by estimating c. In this case, the simplest approach is



(a) claire

(b) football

Fig. 6. Examples of video sequences with 360×288

to calculate $h(c)$ for all possible c. The number of generable codewords c in an image is given as 4^L.

The other approach is to estimate c from each $\sigma(j)$. When $W_H(\sigma(j))$ is an odd, it is satisfied from Eqs.(5) and (6) that $c(j)$ can be decided as $c(j) = \sigma(j) \oplus (001)_2$. On the other hand, when $W_H(\sigma(j))$ is an even, it is impossible to decide $x(j)$. In this case, attackers have to calculate $h(c)$ for 3^{L-Z} codewords c where Z is the number of 3-bit codewords which are generated in step(d) and have $(001)_2$. Therefore, when 3^{L-Z} is larger than 2^{256}, i.e.

$$L - Z > 256 \log_3 2 \simeq 161.28 \tag{13}$$

the key space of the proposed scheme is larger than that of the 256-bits key. As described later, Eq.(13) can be satisfied.

V. Simulation

To evaluate the performance of the proposed scheme, several simulations are conducted

A. Simulation Conditions

We used two video sequences compressed by Kakadu [22] version 6.4. Identification was accomplished by querying each query image respectively. The hash function SHA-256 was used in the simulations. r_{X_i} and r_Q [bpp] represent the coding rate used to generate JPEG 2000 images X_i and Q respectively.

B. Querying Performance

Figure 6 shows examples of the frames of the video sequences. Originally, there were 30 uncompressed consecutive frames for each video sequences. All video frames were compressed under the six coding rates $r_{X_i} = 0.25, 1.0$ [bpp] and $r_Q = 0.5, 0.75, 1.5, 2.0$ [bpp]. As a result, 180 compressed frames were generated from each sequences, and 360 compressed frames were totally used in the simulation. The sets $(\sigma, h(c))$ generated from "claire" compressed under $r_{X_1} = 0.25$ [bpp] were enrolled in database DB_1 and the sets generated from "claire" compressed under $r_{X_2} = 1.0$ [bpp] were enrolled in database DB_2. Similarly, the sets generated from "football" under $r_{X_3} = 0.25$ [bpp] and $r_{X_4} = 1.0$ [bpp] were enrolled in the databases DB_3 and DB_4 respectively. As a query frame, 120 frames generated from the video sequence which was used to enroll in the database under $r_Q = 0.5, 0.75, 1.5, 2.0$ [bpp] were used. Therefore, 120×30 authentication process were performed for each database to evaluate the proposed scheme.

Querying results are shown in Tab.II. The table summarizes the number of true positive (TP), true negative (TN), false positive (FP), false negative (FN) matches. The table also shows true positive rate (TPR) and false positive rate (FPR) defined by

$$TPR = \frac{TP}{TP + FN} \tag{14}$$

$$FPR = \frac{FP}{FP + TN} \tag{15}$$

It is confirmed that there were not any false negative matches for all databases. Compared to "football", FPRs for "claire" increase because it does not include large objective movements between consecutive frames.

Besides, the table shows the minimum value of $L - Z$. It is confirmed that Eq. (13) was satisfied when $L_2 = L_1 - 1$ was chosen. This is because the frames used in the simulations have only 146 code-blocks. If the size of frames is large enough, the condition can be also satisfied even for $L_2 = 1$.

TABLE II
Selection of L_2

L_2	database	TP	TN	FP	FN	TPR[%]	FPR[%]	min$(L-Z)$
1	DB_1	120	3232	248	0	100	7.13	36
	DB_2	60	3539	1	0	100	0.03	75
	DB_3	120	3480	0	0	100	0	27
	DB_4	60	3540	0	0	100	0	51
$L_1 - 1$	DB_1	120	3440	40	0	100	1.15	1081
	DB_2	60	3540	0	0	100	0	3828
	DB_3	120	3480	0	0	100	0	496
	DB_4	60	3540	0	0	100	0	1038

C. Comparison with other methods

The proposed scheme was compared with image hashing-based schemes [18], [19]. The schemes are well known as a robust scheme against lossy coding and they are also applicable to JPEG 2000 images. Therefore, they were applied to the image identification to show that the proposed scheme has a stronger robustness against JPEG 2000 compression than them.

In the image hashing-based schemes, the hamming distances between the hash value of a query image and those of all images in each database are calculated, and then images that have the smallest distance are chosen as the images generated from the same original image as the query, after decompressing all images.

Querying results under the same condition in Sec. V-B are shown in Table III. It is confirmed that $TPR = 100$ was not always satisfied for both methods. It means that the methods provided false negative matches, while there were not any false negative matches in the case of using the proposed scheme. In addition, in terms of FPRs for "claire", the performance of the proposed scheme was better than that of each image hashing-based scheme. Therefore, this results show that the proposed scheme is effective for identifying the frames which do not include large objective movements between consecutive frames.

VI. Conclusion

This paper proposed a codestream level secure identification scheme for JPEG 2000 images. To perform the identification

TABLE III
COMPARISON WITH OTHER METHODS

method	database	TPR[%]	FPR[%]
image hashing [18]	DB_1	60.00	19.63
	DB_2	76.67	13.33
	DB_3	100	0
	DB_4	100	0
image hashing [19]	DB_1	66.67	13.91
	DB_2	91.67	16.58
	DB_3	100	0
	DB_4	100	0

at codestream level, the number of zero-bit-planes which are extracted from JPEG 2000 codestreams was considered. Moreover, to achieve secure identification, the system based on the fuzzy commitment scheme was used. The proposed scheme does not produce any false negative matches in any coding rate. Besides, the identification can be performed without decoding images and storing images. Experimental results show the proposed scheme is effective in terms of true positive matches while keeping the security high.

REFERENCES

[1] C. T. Huang, L. Huang, Z. Qin, H. Yuan, L. Zhou, V. Varadharajan and C. C. J. Kuo, "Survey on securing data storage in the cloud," *APSIPA Trans. on Signal and Information Processing*, vol.3, e7, May 2014.

[2] R. L. Lagendijk, Z. Erkin and M. Barni, "Encrypted signal processing for privacy protection: Conveying the utility of homomorphic encryption and multiparty computation," *IEEE Signal Processing Mag.*, vol 30, no.1, pp.82–105, January 2013.

[3] "Digital Cinema System Specification V1.2," Digital Cinema Initiatives, LLC Technology Committee, March 2008.

[4] O. Watanabe, T.Fukuhara and H. Kiya,"Codestream-Based Identification of JPEG 2000 Images with Different Coding Parameters," *IEICE Trans. Inf. & Sys.*, vol.E95-D, no.4 , pp.1120–1129, 2012.

[5] O. Watanabe, T. Fukuhara and H. KIYA, "Fast Identification of JPEG 2000 Images for Digital Cinema Profiles," Proc. *IEEE Int'l. Conf. on Acoustics, Speech and Signal Processing*, no.IVMSP-L4.6, pp.881–884, May 2011.

[6] O. Watanabe, T. Iida, T. Fukuhara and H. Kiya, "Identification of JPEG 2000 Images in Encrypted Domain for Digital Cinema," in *Proc. IEEE Int'l. Conf. on Image Processing*, vol.2, no.MA.PJ.PJ8, pp.2065–2068, November 2009.

[7] T. Dobashi, O. Watanabe, T. Fukuhara and H. Kiya, "Hash-based Identification of JPEG2000 Images in Encrypted Domain," in *Proc. IEEE Int'l. Symposium on Intelligent Signal Processing and Communication Systems*, no.D2.4, pp.469–472, November 2012.

[8] C. Y. Lin and S. F. Chang, "A robust image authentication method distinguishing JPEG compression from malicious manipulation," *IEEE Trans. on Circuits and Systems for Video Technology*, vol.11, pp.153–168, February 2001.

[9] Z. Fan and R. L.de Queiroz, "Identification of bitmap compression history: JPEG detection and quantizer and estimation," *IEEE Trans. on Image Processing*, vol.12, pp.230–235, February 2003.

[10] D. Edmundson and G. Schaefer, "An overview and evaluation of JPEG compressed domain retrieval techniques," in *Proc. Int'l. Symposium ELMAR-2012*, pp.75–78, September 2012.

[11] F. Arnia, I.Iizuka, M. Fujiyoshi and H.Kiya, "Fast and Robust Identification Methods for JPEG Images with Various Compression Ratios," in *Proc. IEEE Int'l. Conf. on Acoustics, Speech and Signal Processing*, vol.II, May 2006.

[12] K. Iida and H. Kiya, "Secure and Robust Identification Based on Fuzzy Commitment Scheme for JPEG Images,", in *Proc. IEEE Int'l Symposium on Broadband Multimedia Systems and Broadcasting*, pp.1–5, Jun 2016.

[13] K. O. Cheng, N. F. Law and W. C. Siu, "A Fast Approach for Identifying Similar Features in Retrieval of JPEG and JPEG 2000 Images," in *Proc. APSIPA Annual Summit and Conf.*, pp.258–261, October 2009.

[14] A. Chaker, M. Kaaniche, A. Benazza-Benyahia and M. Antonini, "An efficient statistical-based retrieval approach for JPEG2000 compressed Images, " in *Proc. European Signal Processing Conf.*, pp.1830–1834, September 2015.

[15] H. Kobayashi, S. Imaizumi and H. Kiya, "A Robust Identification Scheme for JPEG XR Images with Various Compression Ratios," in *Proc. Pacific Rim Symposium on Image and Video Technology*, November 2015.

[16] K. Iida, H. Kobayashi and H. Kiya, "Secure Identification Based on Fuzzy Commitment Scheme for JPEG XR Images," in *Proc. EURASIP European Signal Processing Conference*, pp. 968–972, August 2016.

[17] A. Swaminathan, Y. Mao and M. Wu, "Robust and secure image hashing," *IEEE Trans. on Information Forensics and Security*, vol.1, pp.215-230, June 2006.

[18] Y. N. Li and P. Wang, "Robust Image Hashing Based on Low-Rank and Sparse Decomposition," in *Proc. IEEE Int'l. Conf. on Acoustics,Speech and Signal Processing*, March 2016.

[19] Y. N. Li, P. Wang and Y. T. Su,"Robust Image Hashing Based on Selective Quaternion Invariance," *IEEE Signal Processing Letters*, vol. 22, October 2015.

[20] A. Juels and M. Wattenberg,"A fuzzy commitment scheme," in *Proc. ACM Conf. on Computer and Communications Security*, pp.28–36, November 1999.

[21] C. Rathgeb and A. Uhl, "A survey on biometric cryptosystems and cancelable biometrics," *EURASIP Journal on Information Security*, 2011:3, September 2011.

[22] "Kakadu software," http://www.kakadusoftware.com/.