Journal of Electrical Engineering 6 (2018) 245-255
doi: 10.17265/2328-2223/2018.05.001

Efficient Watermarking Scheme for Traitor Tracing Encryption Schemes

Kazuto Ogawa and Go Ohtake
Science & Technology Research Laboratories, Japan Broadcasting Corporation, Tokyo 157-8510, Japan

Abstract: In the content distribution services, traitor tracing encryption schemes are useful tools to trace illegal users that distribute content illegally to unauthorized users. However, solo use of these schemes does not necessarily work well and has vulnerability. To complement the property of the traitor tracing schemes, it is effective to use watermarking scheme with them and the watermarking schemes with light load are required. A number of video watermarking methods with light load have been proposed that embed information into compressed video streams. When the compression scheme is MPEG2-Video, its codes are mostly pre-defined using coding tables, and therefore, information can be embedded in the compressed stream by substituting some of the codes. On the other hand, HEVC/H.265 uses arithmetic coding (CABAC) and it is not easy to substitute one code for another in a stream. To deal with this problem, a watermarking scheme for HEVC/H.265 video streams is proposed. It embeds information while the video is being encoded. A broadcasting system incorporating the scheme is also proposed.

Key words: Traitor tracing encryption scheme, watermarking, video compression, HEVC/H.265, arithmetic coding, probability table.

1. Introduction

The distribution environment for audio and video content has dramatically changed, and audio and video content in digital form can now be easily copied and re-distributed illegally. Copyright holders such as broadcasting and movie companies are becoming increasingly concerned about such violations. Copyright protection is a major issue for such applications as pay TV and content distribution services.

Traitor tracing encryption schemes are countermeasures against such violations and a lot of schemes have been proposed [1-6]. However, solo use of the schemes has vulnerability and the use of watermarking scheme is effective in order to complement the traitor tracing encryption scheme [7]. Moreover, the watermarking schemes with light load are required.

Watermarking schemes embed information in an imperceptible form into content. Various watermarking schemes have been proposed [8-14] and some of them have been commercialized [15, 16].

Some of these schemes embed information into compressed video streams [8, 9]. CPU (central processing unit) loads of such schemes are generally lighter than that of the schemes which embeds information in the baseband content. In this sense, the watermarking schemes for compressed streams are efficient. When using MPEG2-Video compression, for example, it is possible to embed the information by substituting some of its code in the coding table with other code. It should be noted that the tables are merely changed. In contrast, the HEVC/H.265 compression method (HEVC) uses arithmetic coding (CABAC) [17], and its probability table is updated often; it cannot use the same embedding scheme used by MPEG2-Video streams. A new watermarking method has to be developed for HEVC streams.

The preliminary version of this paper was presented at the IEEE International Conference on Consumer Electronics 2015 [18]. This paper discusses the details of the watermarking scheme for HEVC considering the use to complement traitor tracing schemes and its application to a broadcasting system, as regards the

Corresponding author: Kazuto Ogawa, Ph.D., research fields: encryption, security, broadcasting technologies.

generation of auxiliary data, content transmission, content reception, information embedding, and watermark detection.

The remainder of this paper is organized as follows: Sect. 2 shows the complementary system for traitor tracing encryption schemes, Sect. 3 presents the watermarking scheme for HEVC encoded streams, Sect. 4 describes an application of the scheme, and Sect. 5 discusses the characteristics of the scheme and its application and clarifies its strong and weak points. This paper concludes with remarks given in Sect. 6.

2. Complementary System for Traitor Tracing Encryption Scheme

Traitor tracing encryption schemes are one of broadcasting encryption schemes. Ogawa et al. [7] then considered the use of traitor tracing encryption schemes in broadcasting system and proposed two complementary systems for the traitor tracing encryption schemes (CTT). One uses a symmetric encryption scheme and the other uses an asymmetric encryption scheme. Fig. 1 shows their symmetric version.

We omit the explanation of details of the system (please see Ref. [7]). Its main characteristics are as follows.

(1) Content is divided into common and private segments ($Content_c$ and $Content_p$).

(2) It uses a watermarking technique and generates two versions of each private segment ($Content_{p0}$ and $Content_{p1}$).

(3) Each receiver can get common segments and one version of each private segment.

(4) It uses three content keys: one (k_{sc}) for common segments and two (k_{s0} and k_{s1}) for private segments.

(5) k_{sc} and k_{s0} are concatenated, and k_{sc} and k_{s1} are concatenated.

(6) Each codeword $d^{(i)} \in \Gamma$ is assigned to receiver (not depicted in Fig. 1).

As you see Fig. 1, the system does not include encoding functions, and hence, it is insufficient as a broadcasting system. We then propose a system that includes encoding functions and inherits the properties of above complementary system. Particularly, we consider an HEVC encoding as its encoding function.

3. Watermarking for HEVC Encoded Stream

For the CTT system, watermarking function is absolutely necessary, and we focus on the watermarking schemes to apply the system.

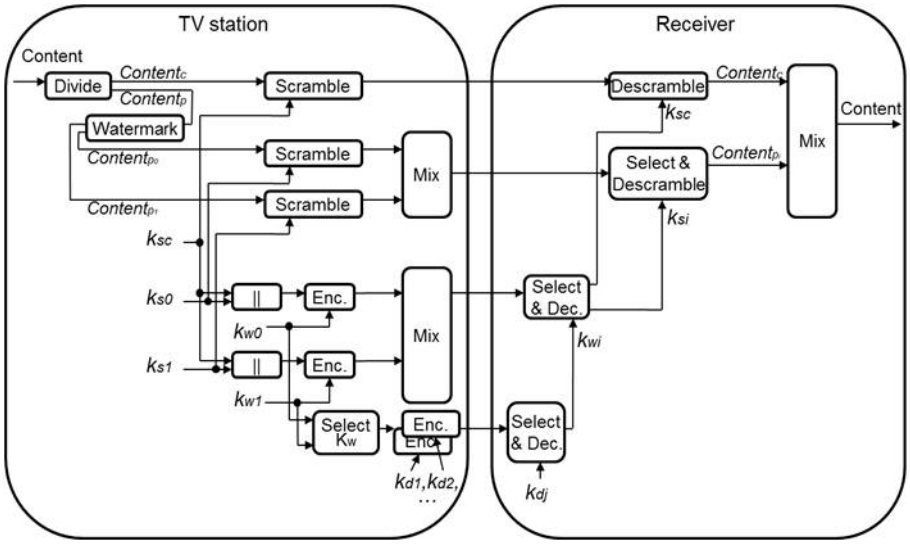


Fig. 1 CTT. $Content_c$ and $Content_p$ denote common and private segments of the content, respectively. $Content_{p0}$, $Content_{p1}$ and $Content_{pi}$ ($i \in \{0,1\}$) denote watermarked content. k_{sc}, k_{s0}, and k_{s1} denote content keys. k_{w0} and k_{w1} denote work keys. k_{dj} for $j = \{1,2,\cdots\}$ denote a device key. || denotes concatenation. "Enc." and "Dec." denote encryption and decryption blocks, respectively.

We then propose a watermarking scheme for HEVC encoded stream. This section describes the scheme's embedding and detection algorithms.

3.1 Embedding Watermarks

Fig. 2 shows the original HEVC encoding procedure. After undergoing a transform, such as a two-dimensional discrete cosine transform (DCT), the transformed data are quantized. The quantized data are then encoded with CABAC. Its details will not be described here. The interested reader can consult HEVC coding [17].

It is almost impossible to embed data into the encoded stream, since the probability table of CABAC for decoding is changed from the one used for encoding by the embedding. The data therefore are embedded before the CABAC process.

Fig. 3 is a block diagram of the embedding algorithm of the proposed scheme showing an example of embedding one bit of information. When the embedded bit is "0", the quantized value is converted into an even value, and when the embedded bit is "1", the quantized value is converted into an odd value. If the information consists of multiple bits (m bits), the process is repeated m times.

This embedding is carried out before the CABAC process, and thus, the probability table for decoding is the same as the one for encoding. The operator that embeds watermarks saves both streams after the CABAC processes, one for an embedded bit "0", the other for an embedded bit "1". It should be noted that two distinct streams continue to be generated after one bit embedding until the next start-code. After the start-code, these streams are identical. When m bits are embedded, two streams are stored for the parts where a certain bit is embedded, and only one stream is stored for the other parts. These stored streams are used when the watermark is detected.

3.2 Detecting Watermarks

There are two distinct algorithms to detect watermarks from a stream, referred to as the target stream, according to the type of the stream. The first algorithm is used when the target stream is the HEVC compressed content that was distributed and has not been modified at all (Algorithm I). The second algorithm is used when the target stream is not the distributed one and has been modified in some way (Algorithm II). These two detection algorithms are described below.

(1) Algorithm I

Let there be a tracer who extracts watermarks from the target stream. He/she picks all parts of the stream in which the bits were embedded. He/she compares all the parts with the two stored streams. There is one identical stream for each part. He/she can decide the bit embedded in each part from the result of the comparison and eventually obtain all bits embedded in the target stream.

If streams are not stored or the tracer cannot access the stored streams, he/she performs CABAC decoding on the target stream. If the quantized value at the watermark embedded position is even, he/she decides the embedded bit is "0", and otherwise, the bit is "1".

(2) Algorithm II

The tracer reconstructs the target content from the

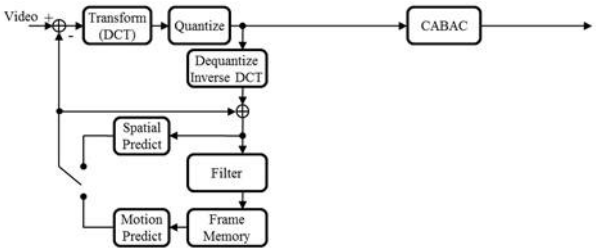


Fig. 2 HEVC encoding procedure.

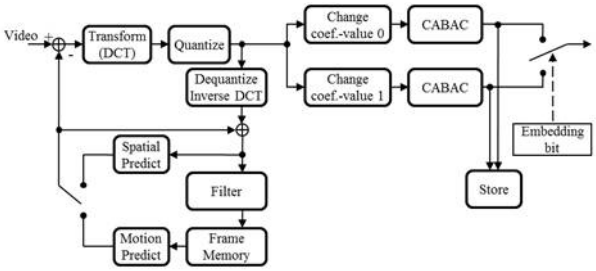


Fig. 3 Proposed watermark embedding method. "Change coef.-value" denotes the block in which a quantized value is changed.

target stream. He/she also reconstructs two versions of the content ("0" embedded content and "1" embedded content) from the two stored streams. He/she picks all parts of the target content in which bits were embedded. He/she compares all the parts of the target content with the parts of the "0" and "1" embedded content at the same positions. If the pixel value in one part of the target content is closer to the part of the "0" embedded content, he/she decides the embedded bit is "0", and "1" otherwise. He/she repeats the same process for all parts and thereby obtains all bits embedded in the target stream.

This algorithm can be used when the target stream was modified, and the computational load of the CPU is larger than that of algorithm I. Naturally, the probability of correct detection from the target stream depends on the robustness of the watermarking schemes to various attacks.

4. Application to Broadcasting System

A broadcasting system that uses the above watermarking scheme for HEVC encoded streams is constructed. It can trace users who illegally redistribute content to unauthorized third parties. That is, it is an application to CTT systems.

4.1 Broadcasting System with Illegal User Tracing Mechanism

It is assumed that the given video content is distributed to multiple users and that a user's or receiver's identifier (U_{id}) is embedded in the compressed content.

Taking the transmission bit rate into consideration, it is impossible to distribute a distinct version of the content to each user or receiver; the broadcaster would not have such a large transmission capacity. Instead, it distributes identical content to all receivers, and each receiver embeds U_{id} in the content [19]. In this case, the CPU of the receiver needs to have high enough performance to embed it in an imperceptible form that can withstand a lot of attacks. However, most receivers do not have such high performance CPU. To enable this for all receivers, the broadcaster generates auxiliary data that help the receivers to embed U_{id} and distributes it with the content. These data are used to reduce the computational load of the CPU.

Only a small part of the content, referred to as the distinct part, is used to embed the watermark information; the other parts, which are referred to as the common part, are not used to embed the watermark. Here, multiple versions of the compressed video stream of the distinct part are generated as auxiliary data, but only one version of the compressed video stream of the common parts is generated. These versions are multiplexed and distributed to the user receivers. Each receiver chooses one version of the multiple versions according to its U_{id} (referred to as the private version). It concatenates the private and common versions and decodes the video content. Fig. 4 is a sketch of the system.

The following subsections describe the generation of auxiliary data, the transmission of content, and the information embedding and detection.

4.2 Generation of Auxiliary Data

The broadcaster generates multiple compressed versions of each distinct part and one compressed version of each common part. The compressed versions of the distinct part are auxiliary data, and the compressed version of the common part is the main data.

Fig. 5 shows the encoding procedure of the proposed system. The broadcaster can embed one or more bits of information into the data by converting certain quantized

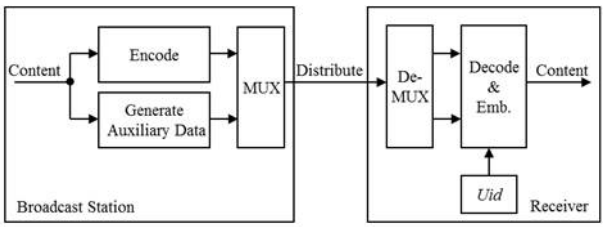


Fig. 4 Proposed broadcasting system. "MUX" and "DeMUX" denote multiplexing and de-multiplexing of input data, respectively.

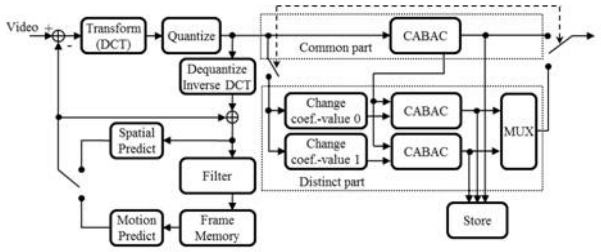


Fig. 5 Encoding procedure.

values into other values. This embedding algorithm is the same as the scheme described in Sect. 3.1, and it can embed one bit of information.

The broadcaster has to generate two compressed versions of the distinct part before transmitting the compressed stream. That is, it has to generate two compressed streams with distinct quantized values at a time and has to perform two CABAC procedures simultaneously. These two versions of the compressed stream become auxiliary data. On the other hand, the CABAC procedure is performed without converting the quantized values in order to generate the main data. These three compressed versions are stored and referred to as "0" embedded, "1" embedded, and original versions.

After the CABAC procedures are performed on the distinct part, two versions of probability table are generated from the quantized values. The tables are not identical, so two compressed streams will be generated. This is impractical because it means twice the transmission bit rate is required.

To solve this problem, after the distinct part is encoded, the broadcaster has to initialize the probability tables and has to make them identical. Taking the transmission capacity into consideration, the initialization had better be performed as soon as possible. Here, the broadcaster inserts a slice-start-code at the beginning of the next CTU (coding tree unit) just after the distinct part, as shown in Fig. 6. In so doing, the probability tables are initialized and become identical. A new common part begins from the CTU and new main data are generated. This common part continues until the next distinct part appears. Until then, only one probability table exists. When the distinct part starts, the probability table for the common part is handed to the CABAC processes of the distinct part.

The compression of the distinct part and the insertion of the start-code are repeated m times, where m is the bit length of $U_{id} = id_1 id_2 id_3 \cdots id_m$. All the generated streams are concatenated and an elementary stream (ES') is constructed. ES' is not the same as the elementary stream (ES) in the original HEVC encoding.

4.3 Content Transmission

The ES' is divided into multiple transport stream (TS) packets or internet protocol (IP) packets and distributed to the receivers. In the following, TS packets are used as an example.

Fig. 7 shows the procedure of generating a TS. There

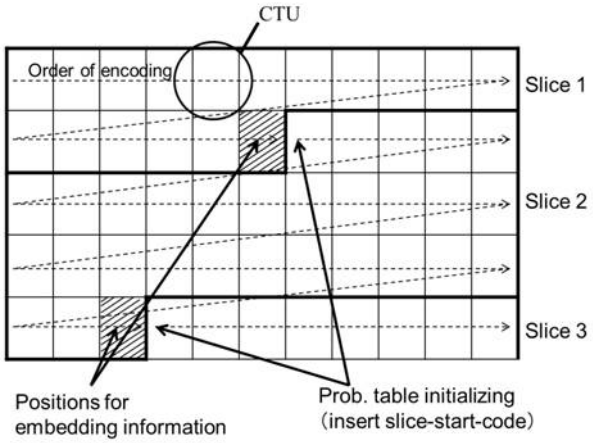


Fig. 6 Positions of embedded information and slice-start codes.

☐ denotes a CTU including only the common part, and hatched ▨ denotes a CTU including the distinct part.

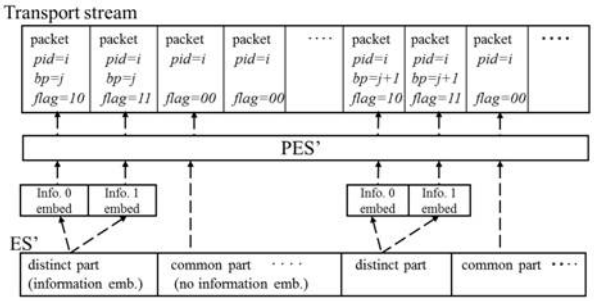


Fig. 7 Generation of transport stream (TS).

"Info." denotes information and "emb." denotes embedding. "PES" denotes a packetized elementary stream, which is an intermediate stream between ES' and TS.

are two versions of the compressed stream for the distinct part, one for an embedded bit "0", the other for an embedded bit "1". Distinct TS packets are used for these two versions. A TS packet for the common part, which starts with a start-code, follows these TS packets.

Each TS packet contains a two-bit flag and a bit-position index. The two-bit flag $flag \in \{00,10,11\}$ denotes the type of TS packet. $flag = 00$ denotes that the packet is for the common part. $flag = 10$ and $flag = 11$ denote packets to embed "0" and "1" information bits, respectively. The bit-position index $bp = \{1, \cdots, m\}$ denotes the position of a bit of U_{id} to be embedded, and the TS packet with the bit-position index $bp = j$ is used to embed a bit id_j. The TS packet also contains an identifier of the program pid.

Multiple TS packets are generated from an ES' with the above procedure. The packets are concatenated to form a TS, which is broadcast.

4.4 Receiving Content and Embedding Watermarks

The receiver receives the broadcasted TS and reconstructs the ES from it. Fig. 8 illustrates the reconstruction. As the reconstruction is being carried out, $U_{id} = id_1 id_2 id_3 \cdots id_m$ is embedded into the stream.

First, the receiver chooses the TS packets that are necessary to reconstruct the ES from the TS. It picks up all the packets that have $flag = 00$. In addition, the receiver picks either the packet that has $flag = 10$ or the packet that has $flag = 11$ transmitted consecutively depending on its U_{id}. More precisely, when the bit-position index bp is equal to k and the k th bit of U_{id} is 0 ($id_k = 0$), the receiver picks up the packet that has $flag = 10$, and when $bp = k$ and $id_k = 1$, the receiver picks up the packet that has $flag = 11$.

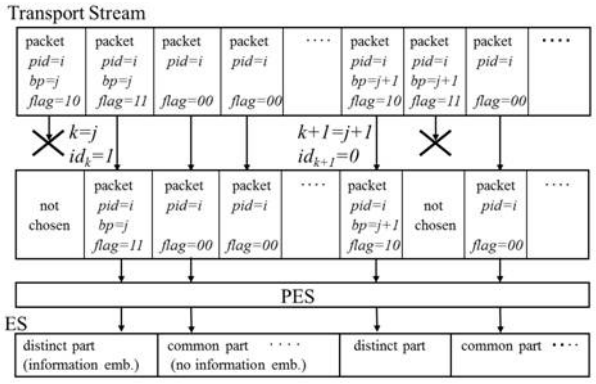


Fig. 8 Reconstruction of ES.

The receiver then removes all the flags and packet headers from the packets, concatenates all the packets it has picked, and reconstructs the ES. It should be noted that the U_{id} has already been embedded in the reconstructed ES. The receiver recovers the video content by decoding the reconstructed ES. That is, it performs the inverse HEVC. Of course, the U_{id} is still embedded in the recovered video content.

4.5 Detection

When illegally redistributed content is found, the embedded information should be detected. A trusted third party, a tracer, does the job. The tracer detects the embedded information and identifies the user or receiver that illegally redistributed the content. In particular, it specifies the U_{id} of the user or receiver by using the detection algorithm described in Sect. 3.2.

- When the HEVC compressed stream, referred to as the target stream, is redistributed as is, the tracer picks all parts of the stream in which the bits were embedded. He/she compares all parts with the two stored streams; i.e., the "0" embedded and "1" embedded versions. There is one identical stream for each part. He/she can decide the bit embedded in the part from the result of the comparison. He/she eventually obtains all bits embedded in the target stream and specifies U_{id}. If a stream is not stored or the tracer cannot access the stored streams, he/she performs CABAC decoding on the target stream. If the quantized value at the watermark embedded position is even, he/she decides the embedded bit is "0"; otherwise, the bit is "1".

• When the decoded content or re-compressed stream is redistributed, the tracer reconstructs the target content from the re-compressed stream. He/she also reconstructs two versions of the content from the two stored streams ("0" embedded and "1" embedded). He/she picks all parts of the target content in which the bits were embedded. He/she compares all parts of the target content with the parts of the "0" and "1" embedded content at the same positions. If the pixel value in one part of the target content is closer to the part of the "0" embedded content, he/she decides the embedded bit is "0", and "1" otherwise. He/she repeats the same process for all parts, obtains all bits embedded in the target stream, and determines the U_{id}.

5. Discussion

Here, the characteristics of the proposed watermarking scheme and broadcasting system are discussed and their strong and weak points are clarified.

5.1 CPU Load of Receivers

As described in Sect. 3.1, there is another method that embeds U_{id} into video content without using any auxiliary data in each receiver. In this case, each receiver has to analyze the content to decide where U_{id} or its individual bits are embedded. This means that firmware or software for embedding information has to be implemented in the receiver. However, the computational load of such firmware or software is not light. For example, it is computationally expensive to perform video analyses involving DCTs.

In the proposed method, the broadcasters perform all of the computationally heavy processes, while the receiver only has to choose TS packets by referring to its identifier when it embeds information. The need for analyzing the video content in the receiver is thus eliminated, and the total load borne by it remains light. It should be noted that making it so that the receiver does not have to perform video content analysis is also beneficial to receiver manufacturers.

5.2 Transmission Bit Rate

Auxiliary data must be simultaneously transmitted with the main data. A large amount of the auxiliary data would not be acceptable from the viewpoint of the transmission bit rate. The increase of transmitted data size is addressed.

The size of the auxiliary data depends on the bit length of U_{id} and on the time of the video content into which U_{id} is embedded. Here, let m (bits) be the bit length of U_{id} and let t (sec) be the time of the video content. In addition, assuming that only the I-frame of HEVC encoded video can be used to embed the data, that the I-frame appears every 15 (frames), and that the frame rate of HEVC encoding is 30 (frames/sec), the number of bits to embed into 1 I-frame is $m/(t \times (30/15)) = m/2t$.

Actual content typically lasts at least several minutes. Here, suppose that $t = 60$ (sec) and m depends on the number of receivers N. Supposing further that N is at most 10,000,000,000, that is more than the world's population, the necessary bit length for m is at most 40 ($m \geq \log_2 10{,}000{,}000{,}000 \approx 36.5$). Therefore, 40 distinct parts would be generated in one minute of content. The distinct part increases by one TS packet, whose size is 188 bytes (1,504 bits). That is, the transmission bit rate increases by $40 \times 1{,}504/60 = 1{,}003$ bps. This value is less than 1/1,000th the current transmission bit rate of video content (several Mbps) and is sufficiently negligible.

5.3 Comparison with Other Watermarking Schemes

The robustness of a watermarking scheme against various attacks has to be considered. Generally speaking, more robustness leads to lower content quality, and content providers/broadcasters have to carefully decide the level of robustness for their services. The robustness of the proposed watermarking scheme will not be discussed in much detail. However, it is possible to lessen the degradation to the quality of content by carefully selecting the positions where the information bits are to be embedded.

The proposed watermarking scheme embeds information bits in a compressed stream. A baseband watermarking scheme, on the other hand, can be used instead. There are a lot of baseband watermarking schemes, and some are robust against HEVC encoding [20]. Such schemes are preferred by providers who want to control the quality of content and at the same time ensure that it is robust against HEVC encoding. The proposed scheme is superior to baseband watermarking schemes in terms of the transmission bit rate and capacity. That is, it is more difficult to control the transmission bit rate when using the baseband watermarking scheme.

To see this, in the baseband watermarking scheme, the provider generates two versions of baseband content, i.e., content in which a bit “0” is embedded and content in which a bit “1” is embedded. These two versions are compressed. The provider decides the range of the distinct part. The compression streams of the distinct part and one compression stream of the common part are transmitted. However, there are a lot of encoding methods for HEVC and they are not identical. For instance, one uses a several-frame feedback loop to obtain a high compression rate, whereas another does not use any frame feedback loop at all. Such differences affect the generated stream. Their compression methods differ in their details. In addition, it is not generally known how these methods divide up a frame of video content into CTUs or coding units (CU), the minimum units for encoding. Therefore, even if it has been decided which CTU is to have some of its pixel values changed and the CTU into which a slice-start-code is to be inserted, only the end point of the distinct part is determined; i.e., the start point is not determined. This is a big issue. When using a baseband watermarking scheme, the provider has to encode the content after embedding a watermark in it, and the result is that the range and size of the distinct part cannot be determined correctly in advance. Hence, it is difficult for the provider to control any increase in the transmission bit rate beforehand.

On the other hand, in the proposed watermarking scheme, only one CTU has two versions of the compressed stream. This makes it easier for the provider to control the transmission bit rate than if they used baseband watermarking.

There are the other watermarking schemes for HEVC. In the proposed scheme, in order to embed one bit of information, the quantized value is changed at most one level. It is possible to change the value multi-level to embed multiple bits. The minimum change is effective not to degrade content much, and thus, multi-level change would not be preferred.

Quantized values are used to embed information in the proposed scheme. Similarly, motion vector values of prediction unit (PU) can be used for the same purpose. When one bit of information is embedded, a motion vector value is changed to another value. However, the change of a motion vector value means the change of pixel values that are used for prediction of pixel values of PU. It sometimes results that quite different pixel values are used for the prediction and results in the degradation of content. It is thus difficult to automatically change the vector value and to correctly control the quality of content.

5.4 Application to CTT system and Enforcement

Fig. 9 illustrates a total CTT system using the above watermarking scheme. The system is based on the broadcasting system that is standardized in Japan [21], which we show in Appendix A. In the Japanese standard, the secure part is implemented in a tamper resistant module such as a smart card and we assume that the CTT system uses a smart card as a secure module.

The system is designed to complement traitor tracing encryption schemes. In addition, it has another characteristic. That is, each receiver must be forced to select one version of the compressed distinct part according to its identifier. In order not to select a different version, encryption schemes have to be used. That takes a role of enforcement on content selection.

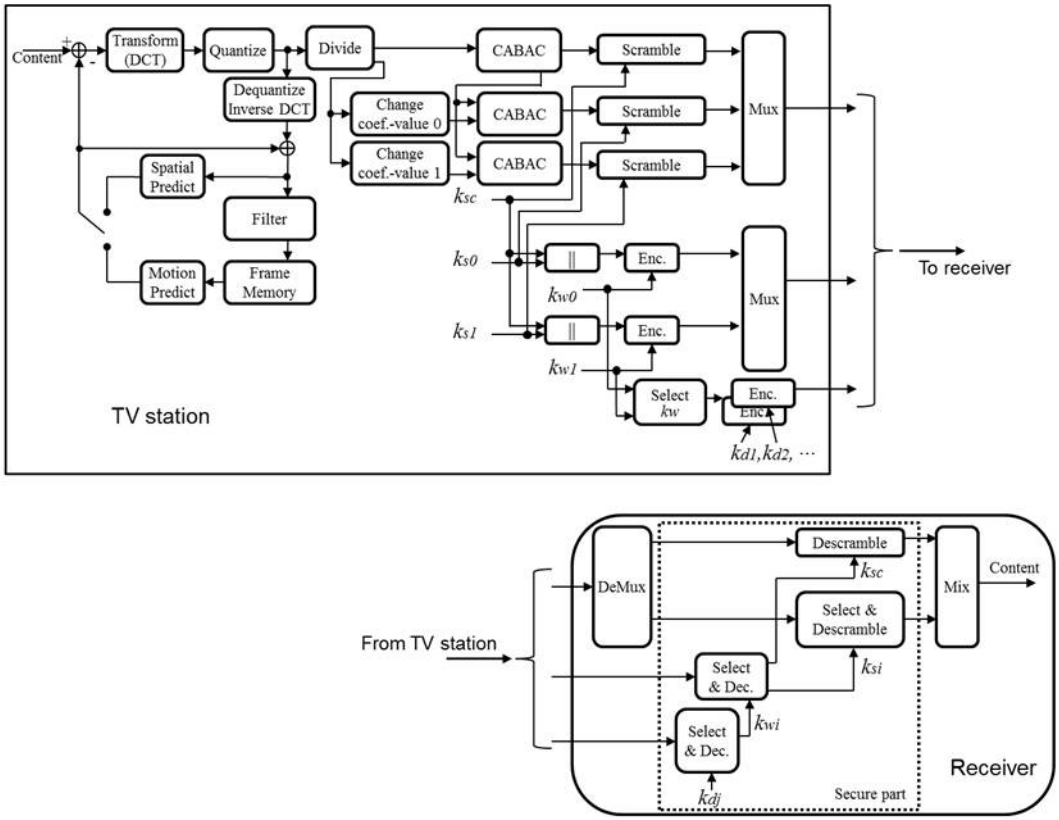


Fig. 9 Total CTT system.

6. Conclusion and Future Works

A watermarking scheme for HEVC encoded stream was proposed and its application to a CTT system was described. The scheme embeds information in the middle of the HEVC encoding process in order to use the same probability table during decoding. With regard to the application system, an actual embedding system that is used for broadcasting services was shown. In addition, it was shown that the computational load of the receivers in the system can be made very light. That is, the CTT system is constructed on the consideration of encoding properties and its loads.

In future, we will have to develop the way how to control quality of the content in the above watermarking scheme. In the above watermarking scheme, we did not mention its quality of watermarked content in details. We assume that the above scheme changes the quantized value by one and that it does not degrade the quality much. However, in order to evaluate the scheme in details and use in practice, we will have to decide the position where the bit is embedded and its level by performing subjective evaluation of its quality. If possible, we would like to use the identical encoding algorithm to that of actually used encoder for the evaluation.

Moreover, the system described here uses the MPEG systems transport stream. MPEG Media Transport (MMT) [22] and MPEG-Dynamic Adaptive Streaming over HTTP (MPEG-DASH) [23] are now standardized and being used. It is possible to employ the proposed watermarking scheme with such transport methods, but the providers may have to modify the above watermarking scheme depending on the transport method they use.

In the proposed system, U_{id} is used as data to be embedded. If the U_{id} is a simple one, its bit length is $m = \log_2 N$. However, a short U_{id} is not sufficient for dealing with collusion attacks, and instead, a

collusion-resistant code [24, 25] for the U_{id} has to be used. As yet however, there is no collusion-resistant code that is practical enough, and the development of a practical one to be able to foil a large number of colluders will have to be waited for.

A broadcasting system as an application of the proposed watermarking scheme was shown. However, some modifications would be to use the proposed scheme in another application. As such, more general schemes would like to be made.

Acknowledgment

We would like to thank Shinichi Sakaida for his valuable comments on HEVC/H.265 encoding/decoding. Owing to his comments, we could improve this paper.

References

[1] Chor, B., Fiat, A., and Naor, M. 1994. "Tracing Traitors." In *Proc. of Crypto'94*, Springer-Verlag, LNCS 839, pp. 252-70.

[2] Kurosawa, K., and Desmedt, Y. 1998. "Optimum Traitor Tracing and Asymmetric Schemes." In *Proc. of Eurocrypt'98*, Springer-Verlag, LNCS 1403, pp.145-57.

[3] Boneh, D., Sahai, A., and Waters, B. 2006. "Fully Collusion Resistant Traitor Tracing with Short Ciphertexts and Private Keys." In *Proc. of Eurocrypt'06*, Springer-Verlag, LNCS 4004, pp. 573-92.

[4] Boneh, D., and Naor, M. 2008. "Traitor Tracing with Constant Size Ciphertext." In *Proc. of ACM CCS'08*, pp. 501-10.

[5] Waters, B. 2011. "Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization." IN *Proc. of PKC'11*, Springer-Verlag, LNCS 6571, pp. 53-70.

[6] Nishimaki, R., Wichs, D., and Zhandry, M. 2016. "Anonymous Traitor Tracing: How to Embed Arbitrary Information in a Key." In *Proc. of Eurocrypt'16*, Springer-Verlag, LNCS 9666, pp. 388-419.

[7] Ogawa, K., Hanaoka, G., and Imai, H. 2017. "How to Make Traitor Tracing Schemes Secure against a Content Comparison Attack in Actual Services." *IEICE Trans. on Fundamentals* E100-A (1): 34-49.

[8] Jordan, F., Kutter, M., and Ebrahimi, T. 1997. "Proposal of a Watermarking Technique for Hiding/Retrieving Data in compressed and Decompressed Video." ISO/IEC Doc. JTC1/SC29/QWG11 MPEG97/M2881.

[9] Hartung, F., and Girod, B. 1998. "Digital Watermarking of Uncompressed and Compressed Video." *Signal Processing (Special Issue on Copyright Protection and Access Control for Multimedia Services)* 66 (3): 283-301.

[10] Sakazawa, S., and Takishima, Y. 2004. "Video Watermarking Method for MPEG Compressed Bitstream." *The Institute of Image Information and Television Engineers Tech. Report*, ME2004-177, 33-8.

[11] Lee, M., Kim, K., and Lee, H. 2009. "Forensic Tracking Watermarking against In-theater Piracy." In *Proc. of Information Hiding'09*, pp. 117-31.

[12] Cao, J., Huang, J., and Ni, J. 2010. "A New Spread-Spectrum Watermarking Scheme to Achieve a Trade-Off between Security and Robustness." In *Proc. of Information Hiding'10*, pp. 262-76.

[13] Du, L., Cao, X., Zhang, M., and Fu, H. 2012. "Blind Robust Watermarking Mechanism Based on Maxima Curvature of 3D Motion Data." In *Proc. of Information Hiding'12*, pp. 110-24.

[14] Zheng, P., and Huang, J. 2012. "Walsh-Hadamard Transform in the Homomorphic Encrypted Domain and Its Application in Image Watermarking." In *Proc. of Information Hiding'12*, pp. 240-54.

[15] Wada, M., Ogawa, K., Fujii, R., Suzuki, M., Magai, K., Itou, H., et al. 2002. "Development of Real-time Video Watermarking Equipment," In *Proc. of Forum on Information Technology'02*, No. 3, J-46, p. 293.

[16] Adobe. "Photoshop/Digimarc." https://helpx.adobe.com/jp/photoshop/using/digimarc-copyright-protection.html.

[17] ISO/IEC. 2013. "Information technology-High Efficiency Coding and Media Delivery in Heterogeneous Environments—Part2: High Efficiency Video Coding." ISO/IEC 23008-2.

[18] Ogawa, K., and Ohtake, G. 2015. "Watermarking for HEVC/H.265 Stream." In *Proc. of IEEE International Conference on Consumer Electronics'15*, pp. 110-1.

[19] Yamada, T., Yoshiura, Y., Echizen, I., Ogawa, K., Murota, I., Ohtake, G., et al. 2003. "Watermarking Application for Broadcast Content Copyright Protection." *The Institute of Image Information and Television Engineers Journal* 57 (9): 1155-67.

[20] Swati, S., Hayat, K., and Shahid, Z. 2014. "A Watermarking Scheme for High Efficiency Video Coding (HEVC)." *PLoS ONE* 9 (8): 313636.

[21] ARIB. 2007. "Conditional Access System Specifications for Digital Broadcasting." http://www.arib.or.jp/english/html/overview/doc/6-STD-B25v5_0-E1.pdf, ARIB STD-B25.

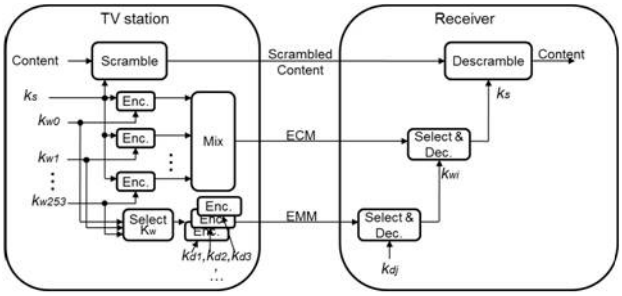
[22] ISO/IEC. 2014. "Information Technology-High Efficiency Coding and Media Delivery in Heterogeneous Environments—Part1: MPEG Media Transport." ISO/IEC 23008-1.

[23] ISO/IEC. 2012. "Information Technology-Dynamic Adaptive Streaming over HTTP (DASH)—Part1: Media Presentation Description and Segment Formats." ISO/IEC 23009-1.

[24] Boneh, D., and Shaw, J. 1995. "Collusion Secure Fingerprinting for Digital Data." In *Proc. of Crypto'95*, pp. 452-65.

[25] Tardos, G. 2003. "Optimal Probabilistic Fingerprinting Code." In *Proc. of ACM Symposium on Theory of Computing'03*, pp. 116-25.

Appendix



App.Fig. 1 Japanese standard: System structure extended for k_w leak source detection.

A.1. Japanese Standard for Broadcasting

App.Fig. 1 illustrates the standard for Japanese broadcasting system [21].

In the standard, a distinct device key k_d is assigned to each model of receivers. The k_d is preset in a secure part of the receiver by a manufacturer. Content is encrypted (scrambled) by using a content key k_s at a TV station. k_s is encrypted using a work key k_w and k_w is encrypted using multiple device keys k_d. The encrypted k_w is included in individual information EMM and the encrypted k_s is included in program information ECM. The encrypted content, ECM, and EMM are multiplexed and transmitted from the TV station to receivers. Each receiver demultiplexes them and gets the encrypted content, ECM, and EMM. The receiver uses the preset k_d, decrypts the encrypted k_w in an EMM, and obtains k_w. The receiver then uses the k_w, decrypts the encrypted k_s in an ECM, and obtains k_s. The receiver finally uses the k_s, decrypts the encrypted content and obtains the plain content. It should be noted that a single work key is common to all receivers.

Authors

Kazuto Ogawa received his B.E. and Ph.D. degrees from the University of Tokyo, Tokyo, Japan in 1987 and 2008, respectively. He joined NHK (Japan Broadcasting Corporation) in 1987. He has mainly engaged in research and development on video image processing systems and digital content rights management systems. He is currently a research engineer of NHK Science and Technology Research Laboratories.

Go Ohtake received the B.E. and M.E. degrees from Tokyo Institute of Technology in 1999 and 2001, respectively. He joined NHK (Japan Broadcasting Corporation) in 2001 and received the Ph.D. degree from Institute of Information Security in 2009. He is currently a research engineer of NHK Science and Technology Research Laboratories. His research interests include public key cryptography and its application for copyright protection and privacy preserving.