ON-LINE MUSIC PROTECTION AND MP3 COMPRESSION

Litao Gang, A.N.Akansu, Mahalingam Ramkumar and †Xuefei Xie

New Jersey Center for Multimedia Research,
ECE Dept., New Jersey Institute of Technology.
†Electrical and Electronic Engineering Dept.,
University of Hong Kong, Hong Kong.

ABSTRACT

In multimedia content protection, encryption is often used to limit the access to authorized users. In this paper, we concentrate on audio content protection. The encryption can be integrated with source encoding to generate the encrypted bitstream or directly scramble the compressed bitstream. We investigate the MP3 format music protection. Several scrambling algorithms in raw data domain and MP3 compression domain are presented for various application scenarios.

1. INTRODUCTION

Multimedia security is an important issue in multimedia email, teleconference, etc. In these applications, only authorized users can get access to the content.

Multimedia protection is different from the general data encryption which involves extensive computation [10] [13]. Two important considerations are *efficiency* and *security*. The former requires real-time operation of the decryption process. This is different from the data scrambling where off-line operation is acceptable. The security requirement is not as rigorous as that in data encryption . Feasible solutions are trade-off between these factors.

Current media encryption algorithms fall into two categories. One integrates scrambling with source coding, viz., to scramble media content before quantization and coding. The other scrambles compressed bitstream. Usually it is desired that the encrypted output is bitstream syntax compatible. Some algorithms have been proposed and applied in video and image scenarios [12] [15].

In this paper, we concentrate on music security. As a case study, we target at MP3 format protection because of its high popularity in on-line music transmission and storage.

In Section 2, we briefly introduce the MP3 compression and its bitstream syntax.

In Section 3, we extend the time-frequency permutation [3] in the MP3 MDCT (Modified Discrete Cosine Transform) domain. One of its advantages is the data manipulation freedom. A major drawback is that permutation would change the original signal statistics, thus making the compression less efficient.

In Section 4, the encryption of the MP3 bitstream is discussed. To avoid confusing the decoder, the encrypted bitstream should comply with the MP3 bitstream syntax. According to different sensitivity requirements, we try to provide different protection levels: 1) *slight protection*, where the encrypted bitstream presents a satisfactory music quality for a casual listener, but not good enough for Hi-Fi reproduction. This could be used to generate different versions for casual users and professionals; 2) *moderate protection*, where the scrambled content is meaningful and the main music features are kept, but with obvious degradation. This could be used for customer evaluation. After test-listening, customers could pay and obtain a decryption key to recover the quality. 3) *maximum protection*, where the music content is completely destroyed thus renders the MP3 bitstream meaningless.

Some conclusions are presented in Section 5.

2. MP3 COMPRESSION

2.1. MP3 Encoding — Overview

The MPEG-1 layer III (MP3) compression is composed of psychoacoustic analysis, subband filtering, MDCT transform, quantization and Huffman encoding.

The polyphase filter bank and windowed Modified Discrete Cosine Transform (MDCT) is employed for subband filtering. The transform length is adaptive to different signal properties based on the perceptual entropy [7].

Human psychoacoustic model is used to shape the quantization noise. One quantization step size (specified by *scale factor*) is applied for coefficients in one

scale factor band. The quantization procedure is composed of two loops — inner loop for bit rate control and outer loop for distortion control.

In the inner loop, the MDCT coefficients are nonlinear quantized and Huffman coded. If the bit budget is larger than the bits available, the step size is increased until the bit consumption is acceptable. Outer loop calculates the quantization distortion in every scale factor band and compares it with the allowed distortion obtained from the psychoacoustic analysis. If it is larger than the allowed distortion, the step size is decreased to reduce the artifact audibility.

The sign and amplitude of coefficients is coded separately. The total MDCT coefficients are divided into 3 regions: big-value region, small-value region and zero region. The big value region (usually at low frequency end) are further divided into 3 sub-regions where different Huffman tables are used. The small-value region is composed of coefficient values of $+1$, -1 or 0. Each codeword represents a pairs of contiguous coefficients in the big-value region or 4 coefficients (quadruple) in the small-value region. The remaining coefficients are implicitly set to zeros (Fig. 1).

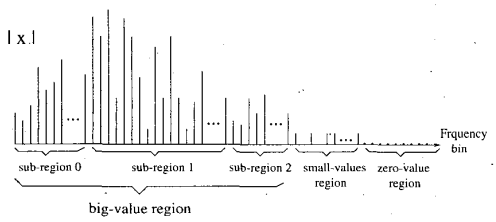

Figure 1: Partitioning of MDCT coefficients

For more detailed description on MP3, refer to [11] [2] and [4].

2.2. MP3 Bitstream Syntax

An MP3 frame is an independent decoding unit [6]. Its **header()** specifies important parameters for decoding operation, for example, bit rate, sampling frequency, coding mode, etc. The encryption should not change this field.

The **audio_data()** field provides the decoding control parameters and MDCT data (**main_data()**). The first half of audio_data() specifies the side information and main_data() just contains the codewords and signs of the MDCT coefficients (Fig. 2).

Although the length of a frame is constant at a given bit rate, bit consumption for samples per frame (1152 samples) is variable. The "bit reservoir" technique permits the current frame to "borrow" bits saved from past frames to absorb the imbalance. The current frame data may locate in previous frames. The location where the main_data() begins is determined by *main_data_begin*, a 9 bit offset value.

```
audio_data()
{
   main_data_begin
          ⋮
   for (gr=0; gr<2; gr++)
     for (ch=0; ch<nch; ch++) {
          part2_3_length[gr][ch]
          big_values[gr][ch]
          global_gain[gr][ch]
          scalefac_compress[gr][ch]
          window_switching_flag[gr][ch]
          if(window_switching_flag[gr][ch])
            block_type[gr][ch]
            mixed_block_flag[gr][ch]
            for (region=0; region<2; region++) {
                table_select[gr][ch][region]
            for (window=0; window<3; window++)
                subblock_gain[gr][ch][region]
          } else {
             for (region=0; region<3; region++)
                table_select[gr][ch][region]
             region0_count[gr][ch]
             region1_count[gr][ch]
          }
          preflag[gr][ch]
          scalefac_scale[gr][ch]
          count1table_select[gr][ch]
   }
   main_data()
}
```

Figure 2: Side Information in MP3 Syntax

3. ENCRYPTION INTEGRATED WITH SOURCE CODING

In this approach, the encryption is performed before quantization and encoding. A widely used signal scrambling method is time-frequency permutation [3]. Fig. 3 shows the block diagram.

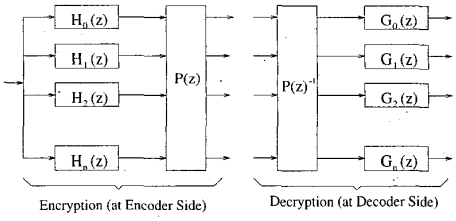

Figure 3: Time-Frequency Permutation

$P(z)$ is a permutation function and its inverse function is $P(z)^{-1}$. Its effectiveness has been proved in practice and can be applied directly in the MP3 MDCT domain. However, the random permutation changes the coefficient distribution property and renders the

Huffman table not optimal. The scrambling also destroys the correlation between contiguous granules which could be used in compression. These result in lower compression rate.

There does not exist an easy solution. A possible remedy to enhance Huffman coding efficiency is to divide the frequency range into several bands, only permute coefficients within a band. This method can keep the coefficient distribution property to some degree, but at the price of compromised security.

In addition, for a stereo signal, the coefficients in one granule can be further permuted between two channels. For most music materials, it is reported that the left and right channels in a stereo source have little correlation [8]. Thus swapping the data in these channels completely destroys the content. This, however, increases encoding (decoding) latency and memory requirement.

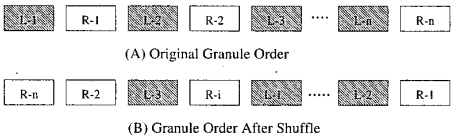

Figure 4: Stereo Signal Granule Shuffle

4. ENCRYPTION IN COMPRESSION DOMAIN

In MP3 bitstream encryption, one requirement is that the encrypted bitstream should be syntax-compatible and the file size should not be changed. We can not simply scramble the bitstream, since that generates an invalid bitstream and confuses decoder.

Generally speaking, the selected Huffman table should not be changed. The minimum unit that can be manipulated is a codeword (of a pair of coefficients in the big-value region or quadruple of 4 coefficients in the small-value region). The encryption can work at following levels: at codeword level, sub-region level and granule level. The scrambling could be one or combination of these operations.

At codeword level, because the coefficient amplitudes and signs are separately encoded, we can permute the codes and (or) flip the signs by a random pattern. While sign flipping can happen on any non-zero coefficient, the permutation should be limited to codes using the same Huffman table for syntax compatibility. For the coefficients whose amplitudes are greater than 15, the *linbits* field can be scrambled without any constraints.

At sub-region level, the sub-regions could be permuted. Respective permutation is also required for related side information parameters, such as code counts and Huffman table index.

At granule level, the granules inside a frame can be reordered randomly. The corresponding parameters, such as *part2_3_length*, etc. should be shuffled for the integrity of the granule.

For different applications, special attention should be taken to meet our degradation requirements.

4.1. Encryption with Slight Degradation

In the MP3 time-frequency decomposition, a fine frequency resolution is applied at low and high frequency bands. At high frequency end, it is not quite necessary. That gives us some room for manipulation.

Most high frequency coefficients are noise component which is an important part in music signal. Some believe that noise energy shape is perceptually significant. It can be described by its DCT coefficients [1] or by a source filter model (for example, a linear predictor (LP) filter).

In contrast, the *Equivalent Rectangular Band (ERB) noise modeling* [5] and *bark band noise modeling* [9] are based on different perceptual assumptions. Both methods keep the noise energy gain without keeping its exact shape. It is believed that human beings do not resolve the fine frequency structure in a noisy band, only a "mixing effect" is felt.

To accurately distinguish noisy bands from non-noisy ones is not an easy job. Not all high frequency coefficients are noisy, some may be the high frequency components of transient signals. In [14], several algorithms are proposed to make a distinction between noisy and non-noisy bands. It is reported that over 80% of the high frequency coefficients are "non-edged". For simplicity, we assume that the frequency components above 5kHz are noisy. Informal listening tests show that it is a reasonable assumption.

The above conclusions on audio signals could be employed in scrambling. For example, we can just flip the signs of the MDCT coefficients over 5kHz. The frequency shape in these bands is unchanged. In addition, we can permute these coefficients within one scale factor band since the noise energy gain is still kept. The modification is almost transparent for a casual listener. If more distortion is permitted, we can even permute and (or) sign-flip lower frequency coefficients. This operation can be further tuned for specific requirements.

4.2. Encryption with Moderate Degradation

To provide more protection, we can extend scrambling to the medium frequency coefficients.

It is believed the frequency amplitude is more important than phase in audio signal. However sign-flipping of the non-noisy coefficients introduces obvious degradation.

The permutation and sign-flipping could be used in this case. To scramble some medium frequency coefficients introduces obvious degradation.

Audio signal spectrum has a wide dynamic range. To keep features of music clips, we skip the large value coefficients, and only manipulate the relatively smaller ones. Experiments reveal that the components under 3kHz are perceptually significant and should not be manipulated much.

4.3. Encryption with Maximum Degradation

To provide maximum protection, we want to completely destroy the audio content while keeping the bitstream syntax. Sign-flipping and codeword permutation could be employed at codeword level.

At sub-region level, the order can be shuffled. Of course, the side information parameters, such as Huffman table index *table_select*, codeword count *region_count* (Fig. 2) etc. should also be permuted accordingly.

The permutation can also happen at higher level. For example, in a stereo signal, two granules in each channels can be shuffled in one frame. To abide by the syntax, we need to change the order of the side information parameters respectively.

5. CONCLUSIONS

In this paper, we investigate the content protection for on-line music. We can encrypt the signal before compression or can directly scramble the compressed bitstream. In compressed domain manipulation, several methods are proposed where human perceptual knowledge is employed to meet different quality requirements. Our experiments with MP3 music signals demonstrate its effectiveness in practice. The same principle could be applied to other audio compression schemes, for instance, MPEG-2 AAC.

6. REFERENCES

[1] ISO/IEC FDIS 14496-3 Sec 2. "Information technology–Coding of audio-visual objects, Part 3:audio, Section 2: Parametric Audio Coding". 1999.

[2] Karlheinz Brandenburg and Gerhard Stoll. "ISO-MPEG-1 Audio: A Generic Standard for Coding of High-Quantity Digital Audio". *J. Audio Eng. Soc.*, 42(10):780–792, Oct. 1994.

[3] Charles D. Creusere and Sanjit K. Mitra. "Efficient image scrambling using polyphase filter banks". *Image Processing, 1994. Proceedings. ICIP-94.*, 2:81–85, 1994.

[4] B. Grill E. Eberlein, H. Popp and J. Herre. "Layer III A Flexible Coding Standard". *Audio Eng. Soc. preprint 3493, 94th Convention, Berlin, Germany*, March 1993.

[5] M. Goodwin. "Adaptive Signal Models: Theory, Algorithms, and Audio Applications". *Ph.D. thesis, University of California, Berkley*, 1997.

[6] ISO/IEC. "IS 11172-3: Coding of moving pictures and associated audio for digital storage media at up to about 1.5Mbits/s". *ISO/IEC*, 1993.

[7] J.D.Johnston. "Transform coding of audio signals using perceptual noise criteria". *IEEE Journal Sel.Areas Comm*, 6:314–323, Feb. 1988.

[8] Ernst Eberlein Jurgen Herre and Karlheinz Brandenburg. "Combined Stereo Coding". *Audio Eng. Soc. preprint 3369, 93rd Convention, Calif, USA*, Oct. 1992.

[9] S. Levine. "Audio Representations for Data Compression and Compressed Domain Processing". *Ph.D. thesis, Stanford University*, 1998.

[10] NIST. "Data Encryption Standard. FIPS PUblication 46-2, 1993".

[11] Davis Pan. "A Tutorial on MPEG/Audio Compression". *IEEE Multimedia Journal*, 1995.

[12] Lintian Qiao and Klara Nahrstedt. "A New Algorithm for MPEG Video Encryption". *International Conference on Imaging Science, Systems, and Technology (CISST'97), Las Vegas*, pages 21–29, 1997.

[13] A. Shamir R.L.Rivest and L. M. Adleman. "A method for Obtaining Digtal Signatures and Public-key Cryptosystems". *Communications of the ACM*, 21(2):120–126, Feb. 1978.

[14] D. Schulz. "Improving Audio Codecs by Noise Substitution". *J. Audio Eng. Soc.*, 44(7/8):593–598, Jul./Aug. 1996.

[15] Changgui Shi and Bharat Bhargava. "A Fast MPEG Video Encryption Algorithm". *Proceedings, ACM Multimedia'98, Bristol, UK*, pages 81–88, 1998.