FREQUENCY-SELECTIVE PARTIAL ENCRYPTION OF COMPRESSED AUDIO

Antonio Servetti[1], *Cristiano Testa*[1], *Juan Carlos De Martin*[2]

[1]Dipartimento di Automatica e Informatica/[2]IEIIT-CNR
Politecnico di Torino
Corso Duca degli Abruzzi, 24 — I-10129 Torino, Italy
E-mail: [servetti|demartin]@polito.it, c.testa@libero.it

ABSTRACT

The widespread adoption of compressed digital audio increasingly demands effective ways to conjugate ease of distribution with content protection. In this paper a low-complexity partial-encryption scheme for MPEG Audio is presented. The aim is to provide listeners with sample-quality audio material that can be upgraded to full-quality by simply acquiring a key and decrypting a few selected bits (1–10% of the total bitstream) of the already available data. Sample-quality is obtained by limiting the frequency content of the signal, and it is achieved without altering the compatibility of the compressed audio bitstream. Audio material partially-encrypted with the proposed scheme can thus be freely distributed for evaluation and then easily unlocked to achieve full quality without any further transmission of audio data. Audio samples are available at http://multimedia.polito.it/icassp03/cryptomp3/.

1. INTRODUCTION

Since ancient times encryption has been applied to protect information. In today's Internet the focus is on securing applications like e-commerce transactions, electronic mail, file transfers. The advent of multimedia streaming and the widespread adoption of compressed digital formats (e.g., MP3), however, poses new security challenges. Arguably the main one is to find ways to conjugate ease of distribution with effective protection of digital video, audio and speech content.

The main idea presented in this paper is that multimedia streams can be subdivided in two parts: a perceptually more relevant fraction to be encrypted, and a remaining part that is less significative and can be left unprotected (or less protected). Work in this direction has recently been proposed for images [1], video [2], speech [3], and audio [4][5][6]. The main advantage of this approach with respect to full encryption of the whole bitstream is its lower complexity (since less bits need to be encrypted), a particularly valuable feature in mobile multimedia scenarios, where security concerns are strong and battery life of paramount importance.

Selective encryption (sometimes referred to also as *partial encryption*), however, is characterized by other significant advantages if compared to regular, full encryption. In particular, selective encryption can be employed not only to achieve the same perceptual effect of full encryption, i.e., complete content protection, but also to deliver a limited, controlled degree of distortion.

This work was supported in part by CERCOM, the Center for Wireless Multimedia Communications, Torino, Italy, http://www.cercom.polito.it.

Limited distortion in this context may be interesting for two main reasons. Limited distortion may be desirable in return for even lower computational complexity; for instance, in the case of speech, loss of intelligibility may be sufficient, instead of complete loss of all perceptual information. Limited distortion, however, may also be welcome in other kind of scenario, that is, when multimedia content is to be distributed in an *upgradable sample mode*. Selective-encryption, in fact, may be employed to create material that, when decoded, generates a limited-quality version of the signal, good enough to be sampled by the prospective buyer, but not enough for serious use. If the user decides to buy the material, a key will be used to unlock the protected part of the bitstream, thus achieving full quality without any transmission of additional data.

We present a novel technique for selective encryption of MPEG Audio Layer III (MP3) bitstreams that generates upgradable sample mode audio material. Sample-quality is obtained by limiting the frequency content of the signal, and it is achieved without altering the compatibility of the compressed audio bitstream. Encryption is applied to a small percentage of the data set, between 1 and 10%.

The paper is organized as follows. In Section 2, we introduce the MP3 compression and its bitstream format. In Section 3, we describe the proposed bitstream-compatible, selective encryption technique for MP3 audio. Results and conclusions are presented in Section 4 and 5 respectively.

2. MP3 AUDIO CODING STANDARD

The MPEG audio coding standard [7] has become a universal standard in fields as diverse as consumer electronics, professional audio processing, telecommunications, and broadcasting [8].

The MPEG-1 layer III (MP3) coding algorithm belongs to the class of perceptual audio coders [9]. The input audio signal is converted into spectral components via a hybrid analysis filter-bank; a polyphase filter bank composed of 32 equal bandwidth filters is employed. To achieve better frequency resolution, each output channel is further subdivided into 18 bands via a windowed Modified Cosine Transform (MDCT). The transform length is adaptive to different signal properties based on perceptual entropy. A human psychoacoustic model is used to shape the quantization noise [10]. Every MDCT coefficient in one scale factor band is multiplied by a common scale factor; subsequently, it is nonlinearly quantized, with the goal of keeping the quantization noise below the masking threshold, and then Huffman encoded. The number of bits for each sub-band and scalefactor is determined on a block-per-block basis. The subband codewords, the scale-

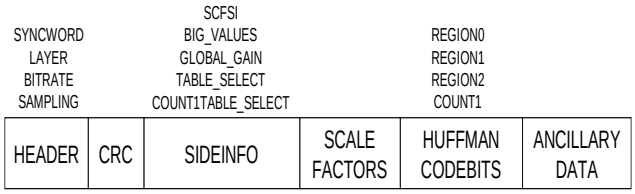


Fig. 1. MPEG-1 Layer III (MP3) frame format.

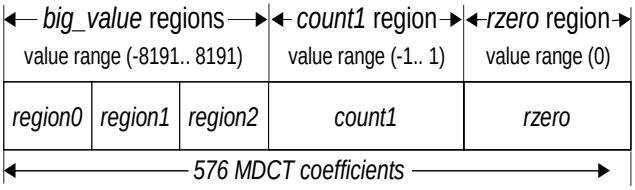


Fig. 2. MP3 bitstream: the partitioning of the MDCT coefficients into five regions.

factor, and the bit-allocation information are multiplexed into one bitstream, together with a header and optional ancillary data.

In the decoder the synthesis filterbank reconstructs blocks of 32 audio output samples from the demultiplexed bitstream.

2.1. MP3 Bitstream Format

MP3 frames, shown in Fig. 1, are independent, that is, each frame contains all the information necessary for its decoding.

Each frame has a header that specifies important decoding parameters, i.e., bit rate, sampling frequency, number of channels. It also contains 12 synchronization bits, 20-bit system information, and an optional 16-bit cyclic redundancy check code.

After the header, the *audio_data* field provides the decoding control parameters and contains side information about bit allocation and scalefactors. The *main_data* field, located at the end of the *audio_data*, contains scalefactors and the Huffman codewords of the MDCT coefficients.

As main information a frame carries a total of 32x36 subband samples corresponding to 1152 PCM audio input samples, subdivided in two granules (or sub-frames), corresponding to about 26 ms at a sampling rate of 44.1 kHz. Subband samples are ordered from low frequencies to high frequencies; variable-length coding is used.

The Huffman coding scheme assumes that large values occur at low spectral frequencies, while low values and zeros occur mainly at the high spectral frequencies, a consequence of the psychoacoustic model applied to the quantization process. Therefore, the 576 spectral lines of each granule are partitioned into five sections [11], as illustrated in Fig. 2.

Starting from the highest frequencies, all the contiguous pairs of zero values are considered to form the *rzero* section. Next, all the contiguous quadruples consisting of values 0, 1 or -1 are assigned to the *count1* section. Finally, the remaining pairs whose absolute values range between 0 and 8,191 (*big_value* pairs) form the last three sections.

To each of the three *big_value* regions is associated a separate Huffman code table. There is a choice of two Huffman tables for the quadruples and a choice of 32 (only 30 are used) Huffman tables for the *big_value* pairs. By individually adapting code tables to sub-regions, coding efficiency is enhanced, while decreasing sensitivity against transmission errors.

The spectral lines are transmitted starting from the lowest frequencies, *big_value* pairs first. The number of *big_value* pairs is sent in the side information section. The number of quadruples is inferred by the encoder when the Huffman data runs out. All the remaining spectral lines are assumed to belong to the *rzero* section. No Huffman encoding is performed on the pairs in the *rzero* section.

For a more detailed description on the ISO MPEG-1 Layer III coding standard refer, e.g., to [12][13].

3. SELECTIVE ENCRYPTION OF MP3 AUDIO

As we described in the previous Secion, MDCT coefficients are partitioned into several frequency regions during Huffman encoding. This spectral subdivision may be exploited to lower the perceptual quality of the compressed signal by low-pass filtering. Limiting the frequency content of audio material is an effective way to generate sample-mode quality, an attractive alternative to the simpler approach of introducing annoying artifacts, such as clicks and pops. The cut-off frequency, moreover, may be modified by increasing or decreasing the number of coefficients that the decoder may decompress, delivering the desired degrees of perceptual quality.

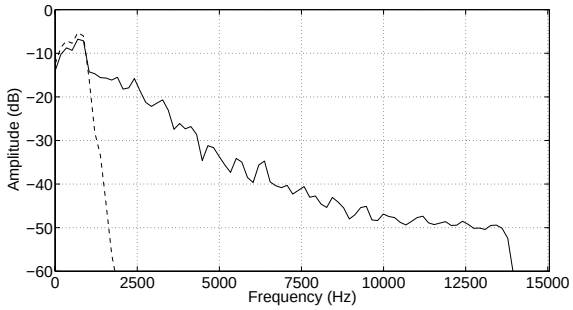
3.1. Low-Pass Filtering in the Compressed Domain

Most of the spectral energy of an audio signal is concentrated in the range from 20 Hz to 14 kHz and usually MP3 encoders map this segment into the *big_value* region. This region is further subdivided in three sub-regions each one coded with a different Huffman table, the one that best fits the sub-region statistics. Most common region boundaries, for a 44.1 kHz sampled signal, are 0-2 kHz for *region0*, 2-5 kHz for *region1*, 5-14 kHz for *region2*. The Huffman table index used for each region is stored in the *table_select* value of the *audio_data* part of the bitstream.

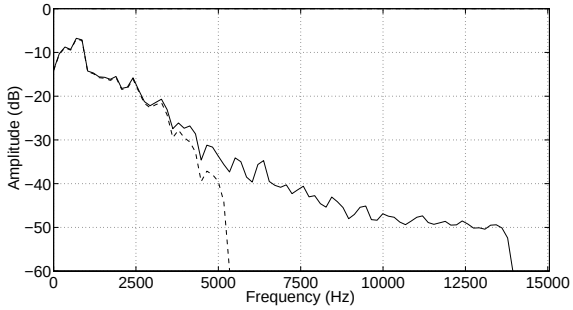
Direct selective encryption of MP3 data in the compressed domain yields very high, uncontrollable distortion at the decoder. Moreover, the modified bitstream can not be properly decoded because Huffman decoding of encrypted MDCT coefficients produces synchronization errors.

It is, however, possible to instruct an MP3 standard decoder not to decode the spectral values of a given frequency region using a so-called private Huffman table index. According to the MPEG-1 standard [7], in fact, among the 32 Huffman table indexes there are two *not-used* values, ‘4’ and ‘14’, that do not correspond to any codeword table. There is no mandatory behavior for the decoder in the case of a *not-used* index, but it usually leads to filling the corresponding portion of the power spectrum with zero value coefficients, i.e., stop-band filtering in the compressed domain.

As first step of the proposed selective-encryption technique, a bitstream modification that permits to obtain a low pass filtered version of the original MP3 signal is introduced. The Huffman table indexes of regions after *region0* are set to the unused value ‘4,’ so that the decoder will skip the decoding of the part of the spectrum that follows region0. Simply changing the table index



(a) 1.5 kHz filter



(b) 5.5 kHz filter

Fig. 3. Frequency content of a sample MP3 audio signal, original versus low-pass filtered: filtering applied (a) after region0, (b) after region1.

of a region, however, is not enough. When the decoder finds a *not_used* index, in fact, it skips the corresponding frequency region and starts decoding the next one using the available data bits. In this case they are not the correct ones because they refer to the skipped region (the right ones are some bytes further on): this causes the incorrect processing of the bitstream and a corrupted frequency content from there on.

The solution to this problem can be found in the MPEG-1 standard. Two steps must be performed to avoid the decoding of spectral values after region0:

- setting the *table_select* index of all regions coming after region0 to *not_used*;
- setting the *big_value* number of line pairs to its maximum , 288, so that the remaining bits are discarded.

The resulting bitstream can be decoded without errors by any MP3 standard decoder and the corresponding audio output will result filtered at approximately 1.5 kHz, if region1 and region2 are modified, or at about 5.5 kHz, if protection is applied only to region2. A frequency analysis is presented in Fig. 3(a) and 3(b) where the filled line is the original spectrum and the dashed one corresponds to the filtered MP3 signal.

To be able to recover the original MP3 bitstream an additional file header is appended at the beginning of the audio track. The header is encrypted to protect its content: the description of the applied protection level, the original *table_select* values and the original *big_value* data. Although, strictly speaking, this does not comply with the MP3 standard, any audio player implemented according to the standard will ignore this information while successfully playing the audio stream.

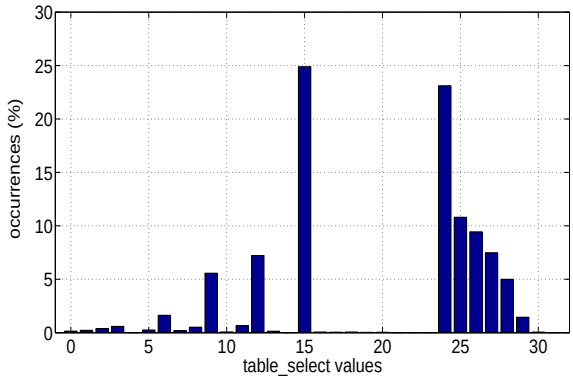


Fig. 4. Statistical analysis of *table_select* values computed on a database of 63 10-second MP3 stereo samples.

The overhead introduced prepending these values can be quantified in 76 bit or 56 bit for each frame, i.e., for a 128 kbit/s 44.1 kHz stereo MP3, a share of only 2.3% or 1.7% of the overall bitstream.

3.2. Protection of Stop-Band MP3 Coefficients

The proposed MP3 data filtering process in the compressed domain produces a compressed bitstream that, for decoders that do not know the decryption key, corresponds to a low-pass version of the original signal. Deciphering, instead, the original *table_select* and *big_value* values, produces the original, full-quality audio stream. This approach, however, is prone to cryptanalysis. Statistical analysis of the encrypted values, in fact, show that MP3 encoders tend to use some *table_select* values more frequently than others. As shown in Fig. 3.2, over a sample database of popular music, about 50% of the Huffman regions is encoded with table numbers ’15’ and ’24’. An attacker could thus replace the modified *table_select* fields with, e.g., ’15’ to obtain an audio signal of acceptable, perhaps even good, quality. The *big_value* parameter is also prone to cryptoanalysis.

The second step of the proposed technique is, therefore, to encrypt part of the data that the attacker would try to decode if the low-pass filtering scheme were to be successfully attacked. Encryption may be applied to a small subset of the compressed bitstream. As explained in [14], in fact, changing a single bit of an Huffman codeword not only modifies its decoded value, but also causes a synchronization error that results in additional errors in the following codewords.

Several encryption patterns were tested and two schemes have been chosen that optimally balance the number of encrypted bits and the introduced distortion. To enable only the decoding of region0 (filter at 1.5 kHz) it is enough to cipher one bit out of twenty in region1: the frequency content of this region is very important and a small modification introduces enough perceptual distortion. The share of protected bits is, therefore, small, only 1.1% of the total bitstream for a 128 kbit/s stereo stream at 44.1 kHz. When the filter is applied at 5.5 kHz (decoding of region0 and region1) the need of encrypted bits is greater: 70-100 bits of region2 have to be ciphered, that is, a share of about 6.3-8.3% of a 128 kbit/s stereo stream at 44.1 kHz.

The combination of low-pass filtering in the compressed domain by modification of the MP3 bitstream (and corresponding

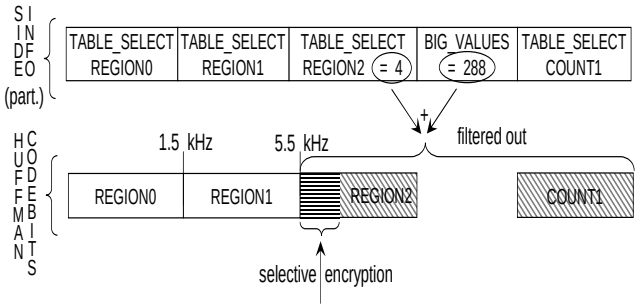


Fig. 5. Proposed MP3 selective encryption applied to region2: the *table_select* value '4' forces the decoder to skip region2. *Big_values* set to 288 artificially extends region2 to the end of the spectrum, so count1 region is not decoded (for simplicity's sake, parameters are not shown in bitstream order).

header protection) and encryption of part of the stop-band coefficients to stymie statistical attacks offers good content protection and the desired *upgradable sample mode* of operation with low-complexity and no modification of the MP3 audio coding standards.

4. RESULTS

The effectiveness of the proposed technique was tested by informal listening tests on a 128 kbit/s stereo signal sampled at 44.1 kHz. Listeners were asked to listen to audio samples filtered at 1.5 kHz and then to the same sample after a cryptanalysis-based attack of the low-pass filtering scheme. All the listeners expressed a clear preference for the former, indicating that no gain is to be expected by attacking the proposed scheme. Listeners were also asked to compare sampled filtered at 5.5 kHz (with the corresponding encryption) and the corresponding cryptoanalyzed sample. The results showed that even in this case, an attacker would not be able to recover any additional perceptual information.

Informal tests showed that low-pass filtering at 5.5 kHz preserves enough audio content for sampling purposes. More formal tests are, however, needed to define the best trade-off between sampling quality, security and overall effectiveness of the proposed scheme.

The cut-off frequency may be changed by the MP3 encoder itself. During the encoding process *big_value* region division in sub-regions can be controlled so that region0 and region1 cover the frequency spectrum to be preserved.

5. CONCLUSIONS

We presented a low-complexity partial-encryption scheme for MPEG Audio. Listeners may experience sample-quality MP3 audio material with standard MPEG audio decoders. If they desire so, the material can be upgraded to full-quality by simply acquiring a key and decrypting a few selected bits (1–10% of the total bitstream) of the already available data. Sample-quality is obtained by low-pass filtering the spectral content of the signal, and it is achieved without altering the compatibility of the compressed audio bitstream. The results of informal listening tests with material low-pass filtered and partially encrypted at 1.5 and 5.5 kHz show that the proposed scheme is robust to attacks based on statistical cryptoanalysis of the low-pass filtering scheme. Audio material partially-encrypted with the proposed scheme can thus be freely distributed for evaluation and then easily unlocked to achieve full quality without any further transmission of audio data.

6. REFERENCES

[1] H. Cheng and X. Li, "Partial Encryption of Compressed Images and Videos," *IEEE Transactions on Signal Processing*, vol. 48, no. 8, pp. 2439–2451, August 2000.

[2] G.A. Spanos and T.B. Maples, "Security for Real-Time MPEG Compressed Video in Distributed Multimedia Applications," in *Proc. IEEE Int. Conf. on Computers and Communications*, March 1996, pp. 72–78.

[3] A. Servetti and J.C. De Martin, "Perception-based selective encryption of G.729 speech," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, May 2002, vol. 1, pp. 621–624.

[4] J. Herre and E. Allamanche, "Compatible scrambling of compressed audio," *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 27–30, October 1999.

[5] L. Gang, A.N. Akansu, M. Ramkumar, and X. Xie, "Online music protection and MP3 compression," in *Proc. of Int. Symposium on Intelligent Multimedia, Video and Speech Processing*, May 2001, pp. 13–16.

[6] N.J. Thorwirth, P. Horvatic, and J. Zhao R. Weis, "Security methods for MP3 music delivery," in *Proc. Asilomar Conf. on Signals, Systems and Computers*, October 2000, vol. 2, pp. 1831–1835.

[7] ISO/IEC, "MPEG-1 coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mb/s," *ISO/IEC 11172*, 1993.

[8] C. Weck, "Unequal error protection for digital sound broadcasting - principle and performance," in *Proc. 94th AES Convention, preprint 3459*, March 1993.

[9] T. Painter and A. Spanias, "Perceptual coding of digital audio," *Proceedings of the IEEE*, vol. 88, no. 4, pp. 451–515, April 2000.

[10] J.D. Johnston, "Transform coding of audio signals using perceptual noise criteria," *IEEE Journal on Selected Areas in Communications*, vol. 6, no. 2, pp. 314–323, February 1998.

[11] S. Shlien, "Guide to MPEG-1 audio standard," *IEEE Transactions on Broadcasting*, vol. 40, no. 4, pp. 206–218, December 1994.

[12] K. Brandenburg and G. Stoll, "ISO-MPEG-1 audio: A generic standard for coding of high-quality digital audio," *Journal of the AES*, vol. 42, no. 10, pp. 780–792, October 1994.

[13] P. Noll, "Mpeg digital audio coding," *IEEE Signal Processing Magazine*, vol. 14, no. 5, pp. 59–81, September 1997.

[14] J.C. Maxted and J.P. Robinson, "Error recovery for variable length codes," *IEEE Transactions on Information Theory*, vol. 31, no. 6, pp. 794–801, November 1985.