2004

Analysis of digital audio compression and digital rights management techniques

Eric Majewicz

An Analysis of Digital Audio Compression and Digital Rights Management Techniques

Eric D. Majewicz
Department of Computer Engineering
Rochester Institute of Technology

Prepared For:
Dr. Chance M. Glenn Sr.
Department of Electrical, Computer, and
Telecommunications Engineering Technology
Rochester Institute of Technology

November 2004

TABLE OF CONTENTS

1. EXECUTIVE SUMMARY

Since the advent of digital audio over 20 years ago, many opportunities have been presented in the proliferation of digital audio content. In the past, the transfer and efficient storage of digital audio has not been feasible due to the large file sizes needed to represent this content. Compression has provided the opportunity to greatly reduce file sizes and make these applications possible. These compression technologies use both lossless compression, which preserves the content exactly by removing redundancies, and lossy compression, which removes perceptually inaudible information to construct a close representation of the content.

This paper starts with an initial discussion on digital audio principles, which is important to understanding advanced digital audio concepts such as compression. The process by which an analog waveform is converted into its digital equivalent is done in three steps: sampling, quantization, and coding. Sampling and quantization take the analog signal, which is a continuous voltage varying with time, and divides it into discrete intervals. Coding is the process by which this information is represented as a series of digital words. A discussion is also presented on the major uncompressed digital audio format for PC, the WAVE file format.

The main focus of this paper is on lossless and lossy compression techniques and their applications. Lossless compression techniques such as Huffman Coding, Arithmetic Coding, and Dictionary-based Coding remove redundancies in the content and code the information as efficiently as possible. These techniques are rarely used alone because of the small compression ratios that they provide as compared to lossy techniques. Lossy compression takes advantage of the fact that the human ear does not perceive all frequencies equally. Instead, critical bands in the ear divide up the frequency spectrum and perceive these bands with different fidelity. Lossy compression techniques that use these principles are discussed, which include subband coding, the psychoacoustic model, and stereo redundancy coding. Applications of these techniques are often referred to as perceptual audio coders, which provide high compression ratios with little loss in audio quality. This paper looks at the following audio coders: MPEG-1 Layer 1,2 and 3, MPEG 2/4 (AAC), Dolby AC-3, and Sony ATRAC. These standards were chosen for evaluation because each of them has an important industry application. MPEG-1 Layer 3, also known as MP3, has been the format used on the internet for transferring and storing music. More recently, AAC has been used for online music services such as Apple iTunes, which is an attractive option because it provides protection of the content as well as compression. Dolby AC-3 is the standard that is used currently used in HDTV, and the ATRAC format is used by Sony's popular MiniDisc device.

This paper concludes with a discussion on Digital Rights Management, which is addressed by many new compression formats. The spread of illegal file sharing across the internet has prompted a strong response from the music industry for formats and standards that can provide both compression and content protection. This paper will look at some of the common techniques that are used for digital rights management such as cryptographic-based protection, digital licensing, and digital watermarking. The applications of these techniques will be discussed by looking at standards such as the Secure Digital Music Initiative (SDMI) and MPEG-4 Intellectual Property Management Protection (IPMP). These standards will be analyzed based on their technologies and the effectiveness in providing sufficient protection.

2. LIST OF FIGURES AND TABLES

3. ABOUT THE AUTHOR

Eric Majewicz is a senior at Rochester Institute of Technology in Rochester, New York. He will graduate in May 2005 with a Bachelors of Science in Computer Engineering. Eric has been working with Dr. Chance Glenn and his company Syncrodyne Systems for the past 6 months on the DYNAMAC digital audio compression algorithm, a new format that achieves compression through chaotic dynamics. Eric has continued to work the DYNAMAC compression application through continued work with Dr. Glenn including an independent study that resulted in this paper, and through research within the Laboratory for Advanced Communications Technology (LACT) at RIT.

4. INTRODUCTION

One of the most brilliant gifts that we have been given as human beings is the ability to hear a wide and diverse range of sounds that contribute to our unique perspective of the world. It has always been a challenge to how we can capture these sounds and record them for others to hear. Early storage mediums included vinyl records, 8-track tape recorders, and the compact cassette introduced by Phillips. In 1982, the digital revolution began with the introduction of the first digital audio 5-inch CD disc, a technology that would merge the consumer music industry with the computer revolution [1]. Since then, the digital format has improved to provide high-quality digital audio to many applications.

In today's high-tech computerized world, digital audio has found a niche is a wide variety of industries. Video games, which used to provide audio in only a series of beeps, now has full soundtracks that are just as important to the overall experience of the game as the visual effects. New cellular phone and PDA technologies allow ring tones to be downloaded directly onto the device, and rich audio content to be transferred to friends. The internet has provided a means of transferring and sharing music with users all over the world. Illegal music sharing on the internet has prompted great response from the recording industry, including pressure for new legislation, and attempts to provide their own legal online music stores. In all of these applications, the compression of large digital audio files has been crucial in transferring and storing digital audio.

As with any technology, new techniques are being researched that can provide better audio quality at higher compression ratios. The current audio coders available today are based on the principle that perceptually inaudible information exists in the content that can be removed with little or no loss in audio quality. As the industry demands higher and higher compression ratios, the limit of how much information can be removed without loss of quality will begin to be reached. The recording industry has also applied pressure to develop audio formats that not only provide high compression, but also a way to protect the intellectual property of the content to curtail the massive amount of illegal music sharing that occurs across the internet. This paper will examine the current technologies that are used to achieve high quality audio compression, and how they attempt to protect this content through digital rights management techniques and standards.

5. DIGITAL AUDIO PRINCIPLES

5.1. Analog vs. Digital Audio

Since the advent of digital audio, much discussion has taken place on whether the analog or digital format is superior for audio. Trends within the last 20 years have shown that digital audio has significantly replaced analog in most industries. This is due to various advantages that digital has over analog sound.

One of the strongest advantages that digital has over analog is it's robustness to noise and error. Analog signals are continuous signals, meaning that there are an infinite amount of voltages that make up the waveform. This allows the signal to become easily corrupted due to external noise. Digital audio on the other hand, is represented in discrete time and voltage intervals using binary. Binary is a robust system, due to the fact that slight variations in voltage will most likely still preserve its output of a 0 or 1. If a digital signal does become corrupted due to a large amounts of noise, error correction techniques such as cyclic redundancy check (CRC) can often be used to correct the result.

Audiophiles and acoustic engineers also favor digital audio because of the ability to easily manipulate the properties of the signal. Using mathematical algorithms, various effects such as reverb can easily be applied. Advanced editing and recording equipment is also much easier to design in the digital domain because digital design uses standardized components. This has lead to audio devices that are smaller with more functionality on the chip, and are cheaper to produce [2].

Digital audio does have some disadvantages however. A conversion process is needed to convert an analog signal to digital. The overall quality of the signal ultimately depends on the quality of the analog-to-digital and digital-to-analog devices used. Only well designed conversion devices can ensure that the fidelity of the signal remains when converted to digital. Another disadvantage is that in order to accurately represent a high quality signal in digital, many samples are needed which results in a large amount of data and large file sizes. In later sections, we will calculate how much data is needed to represent just one second of sound.

5.2. Conversion Process

An analog audio signal can be defined as a continuous voltage that is changing over time within a limited spectrum while a digital audio signal is changing its voltage at certain discrete time intervals[2][3][4]. The process of capturing sound and representing it as a continuous waveform, converting it to its digital equivalent, and converting it back to a continuous form for playing through a loudspeaker is demonstrated in Figure 1. Three steps will be discussed: analog signal capture, analog-to-digital conversion, and digital-to-analog conversion.

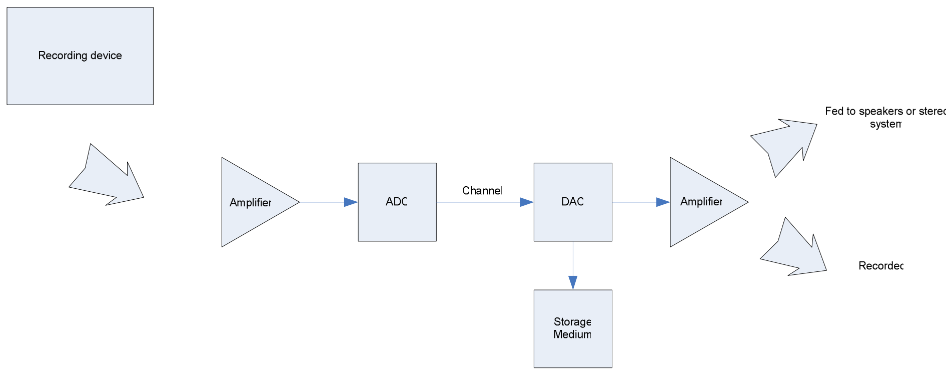

Figure 1: Digital Audio Conversion Process

Analog Signal Capture

Analog sound is captured by a microphone that records variations in sound pressure, or the sound pressure level (SPL). Although there are various types of recording microphones, the simplest example is the moving-coil microphone. This microphone records sound by vibrating a coil of wire that is attached to a diaphragm and suspended in a magnetic field. When pressure is applied to the diaphragm from an air current, the coil vibrates in the magnetic field and generates and electric signal. This signal can then be recorded as a continuous voltage varying with time. Figure 1 shows the recording phase as the first step before digital audio conversion takes place.

Digital-to-Analog Conversion (DAC)

Once a continuous waveform has been captured, the analog signal can be amplified and converted into a digital signal for storage or transmission. This conversion process involves three steps: sampling, quantization, and coding.

The goal of sampling is to discretize a continuous signal across the time domain by taking measurements at fixed time intervals. It is important the time intervals be exactly equivalent to avoid effects that degrade the signal such as delay jitter. As shown in Figure 2, this process breaks up the continuous signal into a finite number of values.

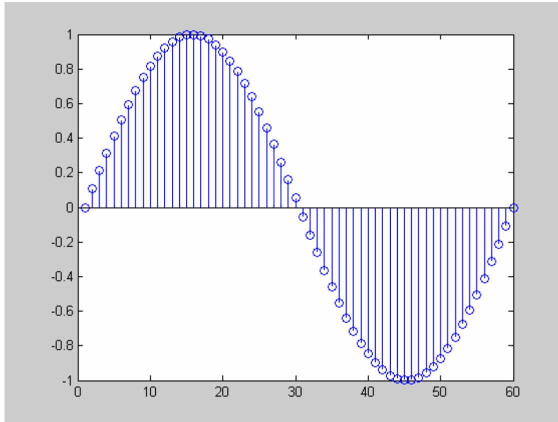

Figure 2: Sampling Process

The choice of a sampling rate is very important to reconstructing a high quality signal. If a sampling rate is chosen that is too low, the signal becomes degraded, loosing high frequency information. If the chosen rate is too high, the excess number of samples will cause high computational load and other difficulties. According to Nyquist's Theorem, the optimal sampling rate is twice the highest frequency of the original signal.

Discretization of voltage also occurs by the process of quantization. Quantization divides the continuous voltages into a discrete number of voltages levels. Figure 3 demonstrates linear quantization, which defines a number of evenly spaced voltage intervals. Non-linear quantization is also possible, and often can achieve better quality.

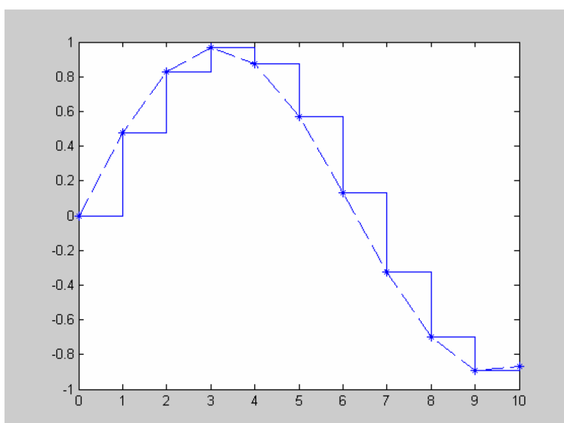

Figure 3: Quantization Process

If an input sample is not assigned a quantization interval that exactly matches its actual voltage, than an error is introduced. This error is called the quantization error or quantization noise. The maximum quantization error for a signal sample is:

$$Q_{errormax} = A/2^n$$

where A is the amplitude of the signal, and n is the number of bits that are used to code each sample. The number of levels used to quantize a signal is 2^n. This number is usually chosen based on an acceptable signal-to-noise ratio, which is defined as

$$SNR = 10\log_{10}(A^2/(A/2^n)^2) = 6.02N + 1.76$$

As shown in this equation, only the number of quantizing levels affects the signal-to-noise ratio and not the amplitude of the signal. For a typical CD-quality audio track, 16-bits are used to encode each level, which produces a signal-to-noise ratio of 96 dB.

The encoding process assigns a unique binary code word to each level, which is n bits in length. For a 16-bit sample, the voltage differential between each level will be:

$$\Delta V = 10 / 65536 = .000153$$

Table 1 demonstrates how binary codes may correspond to a voltage for 16-bit samples:

Voltage (V)	Binary Code
5.000000	0000000000000000
4.999850	0000000000000001
4.999690	0000000000000010
4.999540	0000000000000011

Table 1: Voltages and Corresponding Binary Codes

When this process is completed, it will produce a digital signal that has a bit rate of

Sampling frequency * length of the digital word * no. of channels

Today's CD-quality music defines a 44.1 kHz sampling frequency with 16-bit encoding and 2 (stereo) channels, which produces a bit rate of 44100 * 16 * 2 = 1.41 Mb/s.

Digital-to-analog Conversion (DAC)

In order to reconstruct the original continuous waveform, the encoded samples are converted into a sequence of impulses. This is performed by a digital-to-analog converter which takes each bit and "re-samples" it to reduce the width of the pulses, and than passes the signal through a low-pass filter that has a cut-off frequency which is half the sampling frequency [5]. This produces a fully reconstructed signal that can than be amplified and played through a loudspeaker. It can be shown that with the proper sampling rate chosen using Nyquist's theorem and a sufficient number of quantizing levels, a signal can be fully reconstructed that does not have any similarity to a step-like waveform, disproving any arguments that a digital signal is unable to reproduce high quality audio.

5.3. Digital Audio Standards

WAVE File Format

The WAVE file format is a subset of Microsoft's RIFF specification for the storage of multimedia files. A RIFF file contains a header, and is followed by a series of "chunks". The WAVE file format requires two chunks, the format chunk and the data chunk. The format chunk contains information about the file such as the number of channels, sampling rate, and bytes per sample, while the data chunk contains the actual raw data samples. Although other optional chunks may be present between the format and data chunk, they are not important to understanding WAVE files and will not be discussed.

Represented below in Figure 4 is the structure of a WAVE file with the sizes of each field for the required chunks [6][7]:

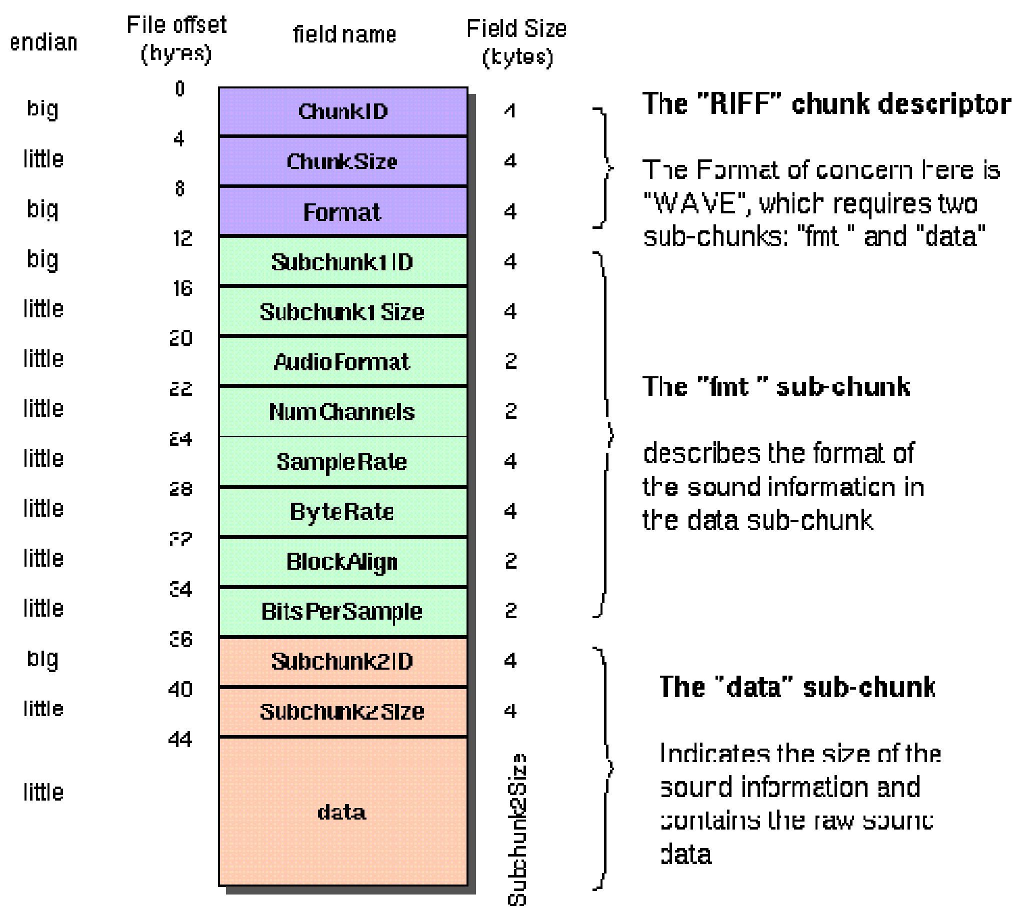

Figure 4: Wave File Format

The RIFF Chunk Descriptor contains header information that is important for identifying the type of file:

ChunkID will contain the letters ”RIFF” to identify that this file uses Microsoft’s RIFF specification.
ChunkSize will contain the total size of the file excluding the *ChunkID* and *ChunkSize*
Format will always contain the letters “WAVE” for WAVE files. Because there are multiple formats that use the RIFF specification, this value will be different depending on the file type.

The “fmt” or sub-chunk describes the format of the sound information in the data chunk:

Subchunk1ID contains the letters “fmt “ to indicate the beginning of the fmt sub-chunk.
Subchunk1Size is the size of the data for this subchunk excluding the *Subchunk1ID* and *Subchunk1Size* values. This value will be 16 for PCM.
AudioFormat will always be 1 for PCM. Other values may be present if compression was used to generate the WAVE file.
NumChannels is the number of channels. This value is either 1 (mono) or 2 (stereo).
SampleRate is the sample rate that was used during the conversion process. This value os 44100 for CD-quality
ByteRate is the *SampleRate* * *NumChannels* * *BitsPerSample*/8
BlockAlign is the number of bytes for one sample including all channels. This value is *NumChannels* * *BitsPerSample*/8.
BitsPerSample is the number of bits used to represent each sample. This value is 16 for CD-quality.

The “data” chunk contains the size of the data followed by the raw data represented as 16-bit integer values:

Subchunk2ID contains the letters “Data”
Subchunk2Size is the number of bytes of raw data contained in the file. This value is *NumSamples* * *NumChannels* * *BitsPerSample*/8
Data is the actual raw data samples represented as 16-bit integer values

6. DIGITAL AUDIO COMPRESSION

6.1. Why Compress?

Emerging digital audio applications in networks, wireless, and multimedia computers face serious shortfalls such as bandwidth limitations, and limited storage capacity. These technologies have created a demand for high quality audio that can be transferred and stored at low bit rates. This creates a need for compression, whose role is to minimize the number of bits needed to retain acceptable quality of the original source signal. Figure 5 demonstrates some of the common applications for the transmission and storage of digital audio content.

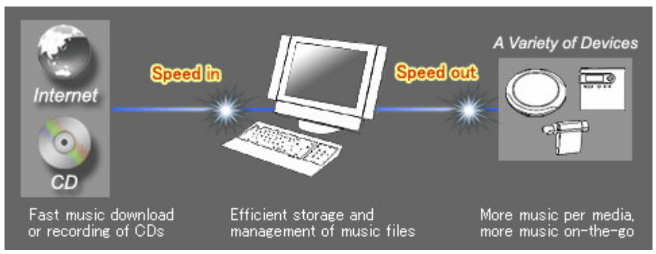

Figure 5: Common Needs for Audio Compression

Fast Music Download
One disadvantage of digital audio music is that it takes a great amount of information to represent a small amount of sound. As demonstrated in the previous section, it takes 1.41 Mb of information to represent one second of music, and an entire song can take up to 50 Mb. It is not feasible or efficient to transfer this uncompressed over the internet, and therefore audio compression helps to reduce the amount of information needed for the digital content and the amount of time needed to download the content.

Efficient Storage and Management on Portable Devices
Portable music devices are becoming the preferred method for storing and listening to digital audio files. As the demand for small portable media devices continues, their storage capacity needs to be maximized to allow digital audio content to be stored as efficiently as possible.

6.2. Design Considerations and Metrics for Audio Coders

When selecting a digital audio encoder for an application, various considerations and metrics need to be taken into account. These considerations include the audio compression rate, audio quality, format of the compressed bitstream, complexity of the encoding and decoding algorithm, and speed of compression and decompression [8].

Compression Ratio
Depending on the algorithms that are used, different coders will possess a different maximum compression ratio. If the maximum possible compression ratio is desired, the best solution will be an audio coder that implements both lossy and lossless techniques.

Sometimes the goal may not be to provide the highest possible compression, but only compression to a certain level. An example is the Sony ATRAC encoder, which provides enough compression only to fit a full music album onto its smaller MiniDiscs. Although additional compression is possible, it would provide additional complexity with no added benefit. The compression ratio of a coder is usually defined as

$$R_{cr} = (s/c) : (c/c) \text{ [dimensionless]}$$

where s is the size of the source file and c is the size of the compression file. Although not as common, the compression ratio may also be defined as a percentage such as

$$R_{cp} = (1 - 1/(s/c)) + 100 \text{ [\%]}$$

Audio Quality

Most audio coders use lossy compression as they are willing to sacrifice small degradations in audio quality for a higher compression ratio. Lossless compression is used either for data compression, where it is important that all the data is preserved, or in conjunction with a lossy techniques to provide additional compression. There is no need to determine audio quality for lossless compression as it perfectly preserves the signal. Quality of a lossy compression signal can be determined either objectively or subjectively. Objective metrics use the signal-to-noise ratio which can also be defined as [9]

$$SNR = \sum_{i=1}^{N}(x_i)^2 / \sum_{i=1}^{N}(x_i - y_i)^2$$

where x is the original source signal and y is the corresponding reconstructed signal. Audio quality can also be determined subjectively using the mean opinion score (MOS), which is a formal subjective quality evaluation scheme that involves a category-judgment technique based on presenting phonetically-balanced Harvard sentences to a selected listening audience. The audience evaluates the difference between the original signal and it's reconstruction on an integer scale ranging from 1 (very annoying) to 5 (imperceptible).

Compressed Bitstream Format

Often it is important that the format complies with a certain standard to guarantee interoperability between devices. For this reason, the MPEG format is a popular choice because of its wide acceptance in the industry. If interoperability is not an issue, than compression format does not need to be taken into consideration. A good example is again the Sony MiniDisc devices, which need only to be able to understand the format contained on a Sony MiniDisc.

Complexity and Delay of the Encoder/Decoder

Complexity and speed are usually directly related for audio encoders: those that achieve higher compression ratios require more complex algorithms and produce higher delays during compression and decompression. Examples of this are the various layers of the

MPEG-1 family. While each successive layer provides additional compression, they also require more complexity and therefore more delay. Complexity metrics can be determined using arithmetic processing measured in either millions of instructions per second (MIPS) or millions of operations per seconds (MOPS). If a high-level language is used for implementation, algorithmic complexity can be measured using O(n) notation. If the audio coder is hardware based, the chip complexity can be determined by either the number of transistors or gates.

6.3. Lossless Compression Techniques

Lossless compression techniques provide for exact reconstruction of the original signal. Lossless compression works by removing redundancies within the data and than coding the resulting signal with an efficient coding scheme [10]. While lossless compression does provide for perfect reconstruction of the original waveform, it is usually not preferred as the sole compression method for digital audio because of its limited compression ratio. It's rare to see a lossless compression algorithm that can provide a compression ratio above 3:1. Instead, it is often applied in conjunction with a lossy compression scheme to provide additional compression. Three techniques that are often used for lossless audio compression are Huffman Coding, Arithmetic Coding, and Dictionary-based Coding.

Huffman Coding

Huffman coding has been shown to be one of the most efficient and simple variable-length coding techniques used in high-speed data compression applications. Two important principles in Huffman coding is that no code is a prefix of another code which allows for a unique way to decode each word, and that no information is needed as a delimiter between codes. Another property is that with codes that have the same code length, any re-ordering of the codes in the Huffman table does not change the coding effectiveness (entropy) [10][11].

The following five steps are used to create a Huffman Tree which efficiently assigns codes to each symbol:

1. Label each node with one of the source letters and it's corresponding probability
2. Merge the nodes labeled with the two smallest probabilities by making them descendants of a common ancestor.
3. Label their ancestor with the sum of their probabilities.
4. Continue merging the nodes with the two smallest probabilities together until All nodes have been merged.
5. Assign a 0 to each left branch and a 1 to each right branch

Figure 6 depicts visually the algorithm for constructing a Huffman Tree:

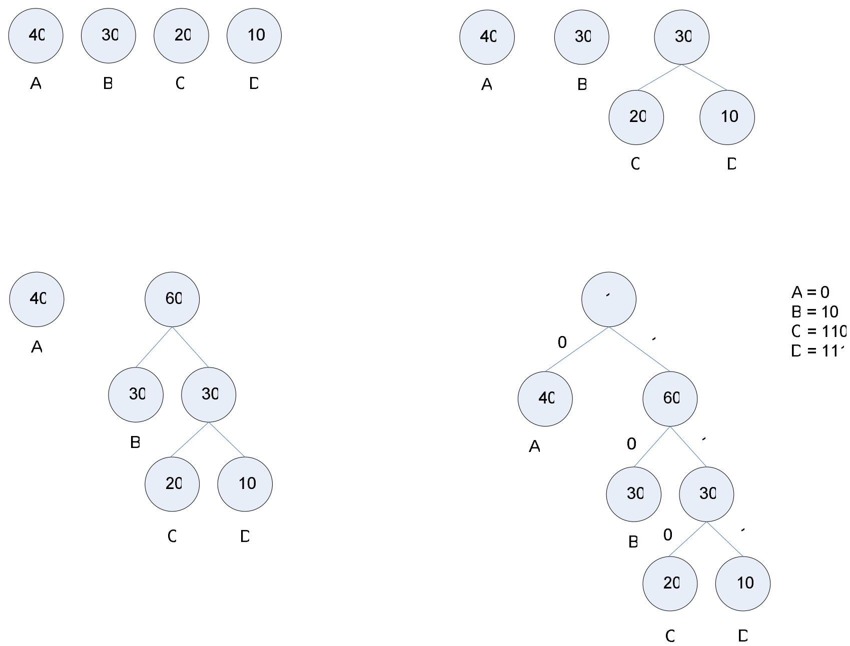

Figure 6: Construction of a Huffman Tree

The effectiveness of this technique can be measured my looking at the overall entropy of the message after being assigned Huffman Codes:

$$l = (.4*1) + (.3*2) + (.2*3) + (.1*3) = 1.9 \text{ bits/symbol}$$

Arithmetic Coding

Arithmetic coding is another lossless technique that has quickly gained acceptance because of its speed, low storage requirements, and effectiveness. Although it is similar to Huffman Coding, it achieves superior performance due to the fact that Huffman Coding asserts that each symbol must translate into an integral number of bits in the encoding, while arithmetic coding represents a message between an interval of real numbers from 0 to 1. As the message grows, the interval needed to represent the message becomes smaller and the number of bits needed to specify it grows [12][13].

To understand how the arithmetic coding algorithm works, consider the example in Table 2 that contains the symbol alphabet {a, e, i, o, u, !}.

Symbol	Probability	Range
A	.2	[0, 0.2)
E	.3	[0.2, 0.5)
I	.1	[0.5, 0.6)
O	.2	[0.6, 0.8)
U	.1	[0.8, 0.9)
!	.1	[0.9, 1.0)

Table 2: Arithmetic Coding Symbols and Ranges

Suppose that the message aeii! is to be transmitted and the decoder knows that the initial range is [0,1). After seeing the symbol 'e', the encoder narrows the range to [0.2, 0.5]. The second symbol 'a' narrows the range down to one-fifth of that range, since 'a' has been given a range of [0, 0.2). This produces [0.2, 0.26), since the previous range was .3 units long and one-fifth of that is .06. After going through each symbol, the range is narrowed down to:

	[0, 1)
e	[0.2, 0.5)
a	[0.2, 0.26)
i	[0.23, 0.236)
i	[0.233, 0.2336)
!	[0.23354, 0.2336)

In order to decode this message, only the range [0.23354, 0.2336) is needed. When the decoder sees this message, it immediately knows that 'e' is the first symbol as the message range fits within the range of 'e'. Now it can use the same methodology as the encoder to retrieve the symbols:

initially	[0,1)
after e	[0.2, 0.5) ...

this makes it clear that the next message is 'a', since this will produce the range [0.2, 0.26). Proceeding in a reverse engineering manner, the decoder can identify the entire message. Using the arithmetic coding technique, the entropy of the system is:

$$-\log_2 0.3 - \log_2 0.2 - \log_2 0.1 - \log_2 0.1 - \log_2 0.1 = 4.22 \text{ bits/symbol}$$

Dictionary-based Coding

Dictionary-based coding is one of the most commonly used techniques used in lossless compression because it provides excellent compression with a low degree of complexity [14]. Most Dictionary-based coders are based on the Limpel-Zev family, such as LZ77 and LZ78. Dictionary-based coding techniques work by replacing frequently occurring strings with a code word which corresponds to an entry in the dictionary. The more frequently occurring strings are assigned shorter code words. There are two different approaches to dictionary-based coding using a static dictionary or an adaptive dictionary.

A static dictionary is most appropriate to use when considerable prior knowledge about the source is available. One of the most popular coding techniques for static dictionaries is diagram coding. With diagram coding, all letters of the source alphabet are included in the dictionary as well as common pairs of letters, called diagrams. An example for a diagram coding technique is shown below:

abracadabra

Code	Entry	Code	Entry

000	a	100	r
001	b	101	ab
010	c	110	ac
011	d	111	ad

Table 3: Sample Static Dictionary

The encoder reads the first two characters and checks to see if this pair is in the dictionary. It finds the pair and encodes the pair using the codeword 101. For the next two characters, no pair exists in the dictionary to it encodes the character r and a using their respective code words.

The other approach is using an adaptive dictionary. Although there are many coding schemes that use the adaptive dictionary technique, one of the most common ones is the LZW algorithm. Take the following example:

wabbawabba

The encoder first encounters the w. Since this is already in the dictionary, it concatenates it with the next character, forming wa. Since wa is not already in the dictionary, it encodes w with it's index value and adds wa to the dictionary. Once the encoder arrives at the second w, it sees that wa is already in the dictionary and moves onto wab. Since wab is not in the dictionary it encodes wa with it's index value and adds wab. This process is repeated until a comprehensive dictionary is generated for the input source. As this process continues, longer codewords are added demonstrating that more of the structure in the sequence is being captured.

6.4. Lossy Compression

Lossy compression works by exploiting the irrelevancy contained within an audio signal largely using psychoacoustic principles. Lossy compression is preferred to lossless because of the greater compression ratios that can be obtained. When considering lossy compression, both the rate of compression and the level of distortion need to be taken into account. Distortion tells how close the reproduced signal is from the original signal before compression. Three compression techniques are often used during lossy compression: subband coding, bit allocation using the human psychoacoustic model, and joint stereo coding.

Subband Coding

Subband coding attempts to model the critical bands of the human ear by splitting the audio signal into a number of frequency bands and grouping them based on where their loudest spectral component exists. The bandsplitting process is complex and requires a lot of computation. One bandsplitting method often used for this is quadrature mirror filtering (QMF), another technique that is used by newer perceptual coders is modified discrete cosine transform (MDCT).

The number of subbands depends on what other compression tools are to be combined with the sub-band coding [16]. If it intended to optimize compression based on auditory masking, the subbands should be narrower than the critical bands of the year, and therefore a large number is required. There are some disadvantages to using a large number of subbands such as complexity and the coding delay.

Subband coding is used in conjunction with psychoacoustic signal analysis to allocate an appropriate number of bits for quantization. The approach is to allocate a just-sufficient number of bits to mask quantization noise while simultaneously satisfying some bit-rate constraints.

Psychoacoustic Model
The psychoacoustic model is a critical part of perceptual audio coding that exploits masking properties of the human auditory system. The psychoacoustic model analyzes signal content and combines induced masking curves to determine what information below the masking threshold that is perceptually inaudible and should be removed. Two types of masking can occur: simultaneous masking and temporal masking.

Simultaneous masking is a frequency domain phenomenon where a low-level signal (the maskee) can be made in-audible (masked) by a simultaneously occurring stronger signal (the masker) as long as each of them are close enough in frequency [17]. A masking threshold can be measured and low-level signals below this threshold will be inaudible. Once all the individual masking thresholds are calculated, they are combined to form a global masking threshold, which is also referred to as the just noticeable distortion. Figure 7 shows the affects if simultaneous masking. Without a masker, a signal is inaudible if its sound pressure level is below the threshold of quite for a specific frequency, also shown in Figure 7. Finally, the global signal-to-mask ratio (SMR) can be determined as a ratio of the maximum of the signal power and global masking threshold. [18]

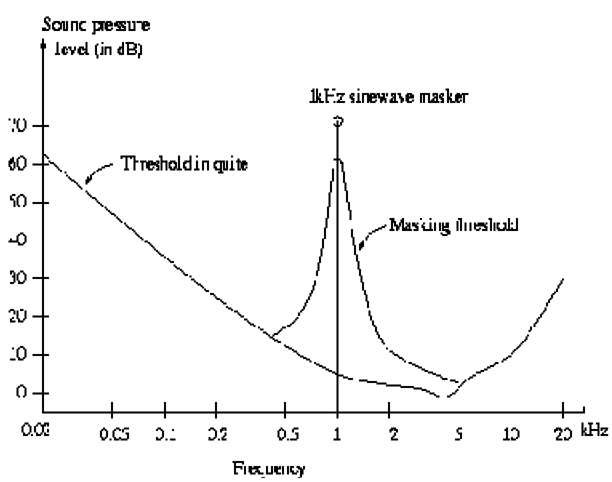

Figure 7: Simultaneous Masking

The distance between the masking tone and the minimum masking threshold is referred to as the signal-to-mask ratio (SMR). Within a critical band, a signal will be inaudible as long as the masking tone is above the masking threshold. Figure 8 shows the masking threshold and the signal-to-mask ratio.

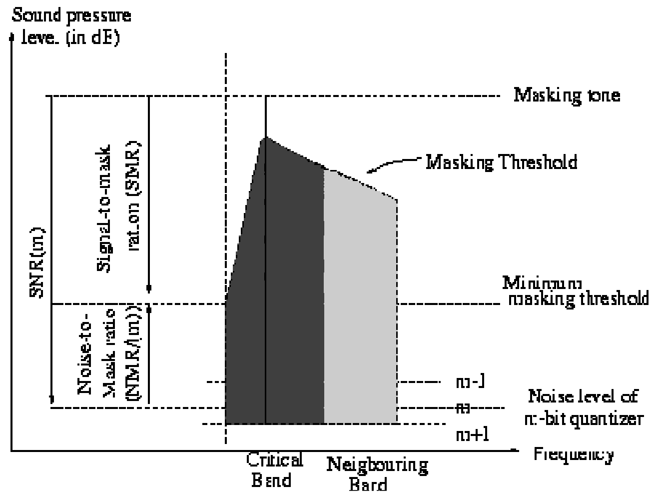

Figure 8: Masking Threshold and SMR

Temporal occurs in the time-domain when two signals appear within a small interval of time. The stronger sound will mask the weaker one, even if the maskee precedes the masker Temporal masking effects occur before and after a masking signal has been switched on and off. Figure 9 shows the effects of temporal masking. Note that the duration of pre-masking is less than one-tenth that of the post-masking, which is usually about 50-200 msec [19].

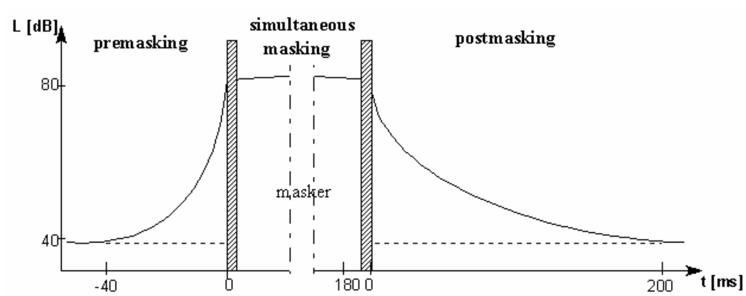

Figure 9: Temporal Masking

Once all masking principles have been applied, a signal-to-mask ratio (SMR) is determined. The signal-to-mask ratio determines the amount of noise which can be imperceptibly introduced into each critical band. The SMR is than used to adjust the bit allocation in each of the sub-bands to determine the optimal coding for each band.

Stereo Redundancy Coding

Stereo redundancy coding exploits another perceptual weakness of the ear. Psychoacoustic results show that in the critical bands that contain frequencies approximately over 2 Khz, the perception of stereo imaging is based more on the temporal envelope rather than the temporal fine structure [15]. There are two types of stereo redundancy coding: intensity stereo coding and MS (middle/side) stereo coding. For intensity stereo coding, the sum of the two channels is coded rather than coding the two channels independently. The decoder than decodes the signal based on independent left and right scale factors. For MS stereo coding, the left and right channels are encoded as middle (sum of left and right) and side (difference of left and right). In this mode, the encoder uses special techniques to further compress the side channel.

6.5. Audio Compression Standards

As noted in the previous section, most standards that have arisen for digital audio compression rely on both lossy and lossless compression methods. The most popular audio compression standard used in many consumer applications is the Moving Picture Experts Group (MPEG) standard. The MPEG-1 standard includes three different coding schemes: Layer 1, Layer 2, and Layer 3. Each is more complex than the next and provides an additional amount of compression. Each Layer N decoder is backwards compatible with previous layers, as is defined by the MPEG standard. Table 2 demonstrates the achievable bit rates and compression ratios for each layer [20]

MPEG-1/Audio Coding	Approximate bit rates	Compression Factor
Layer I	384 kb/s	4
Layer II	192 kb/s	8
Layer III	128 kb/s*	12

* Average bit rate, uses variable bit rate coding

Table 4: MPEG-1 Bit Rates and Compression Factors

Since the great success of the MPEG-1 standard, improvements have been made which are incorporated into the MPEG-2 Advanced Audio Coding (AAC) and MPEG-4 audio object coding standard. MPEG-4 is an attractive option for online music stores because of its implementation of a standard for digital rights management. This will be discussed in detail in later sections.

Aside from the MPEG family of coders, other perceptual audio coders have been developed that have been adopted in certain industries. Dolby AC-3 is the accepted standard for HDTV while Sony uses the ATRAC coder it's MiniDisc.

MPEG-1 Layer 1/2

The MPEG-1 uses a very similar process to implement its Layer 1 and Layer 2 standard. Figure 10 shows the block diagram for these encoders.

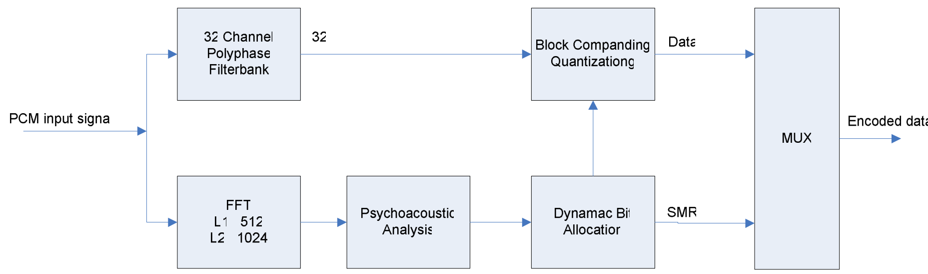

Figure 10: MPEG-1 Layer 1/2 Encoder

Although each layer provides additional complexity and compression ratios, the standard uses the same general process of a perceptual audio encoder. The process for perceptual audio encoding include the filterbank, perceptual model and bit allocation, and finally quantization and coding.

Filterbank

The filter bank for MPEG-1 Layer 1 and Layer 2 is a simple polyphase filter bank that uses a bank of 32 filters to split the input signal into 32 bands, each with a bandwidth of $f_s/64$, where F_s is the sampling frequency. The goal is to model the process that occurs in the ear by the cochlea which splits up a signal based on established critical bands. Sampling frequencies of 32,000 samples per second, 44100 samples per second, and 48000 samples second are allowed [2].

Perceptual Model

The perceptual model, or psychoacoustic model, applies principles of the human hearing system such as absolute hearing threshold, critical band frequency analysis, and auditory masking, to determine the optimum bit allocation for each subband. Figure 10 shows one key difference between the Layer 1 and Layer 2 encoder: Layer 1 uses a 512-point FFT for time-to-frequency mapping of the signal while Layer 2 uses a 1024-point FFT to achieve finer granularity. The output of the perceptual model consists of values for the masking threshold or just allowable noise for each subband. If the quantization noise can be kept below the masking threshold for each subband, than the compression result should be indistinguishable from the original signal.

Quantization and Coding

The spectral components and re-quantized and coded so that the noise level is just below the masking threshold. Unlike a PCM block, a coded block contains many different wordlengths that may vary from one sub-band to the next. Since samples in the same blocks will have different allocations, the decoder needs to be told what bit allocations were used. A bitstream formatter is used to assemble the bitstream [17]. The format for a Layer 1 frame is shown in Figure 11.

Figure 11: MPEG-1 Layer 1 Frame

Another variation between the Layer 1 and Layer 2 encoder is that Layer 2 includes an additional value in each frame, shown as SCFSI in Figure 12. Each Layer 2 frame consists of 36 different subband samples, which generates 3 scale factors per frame [2]. The data rate for these scale factors can be reduced by taking advantage of redundancies. The three subband scale factors are analyzed for patterns. The pattern will decide whether 1, 2, or all three scale factors are required. This information will be communicated by the scale factor select information data word of 2 bits (SCFCI).

Header	CRC	Bit allocation	S C F S	Scale Factors	Sub-band samples	Aux Data

Figure 12: MPEG-1 Layer 2 Frame

MPEG-1 Layer 3

MPEG-1 Layer 3, more commonly known as MP3, is the most complex layer, and also provides the highest level of compression. The block diagram is shown below in Figure 13.

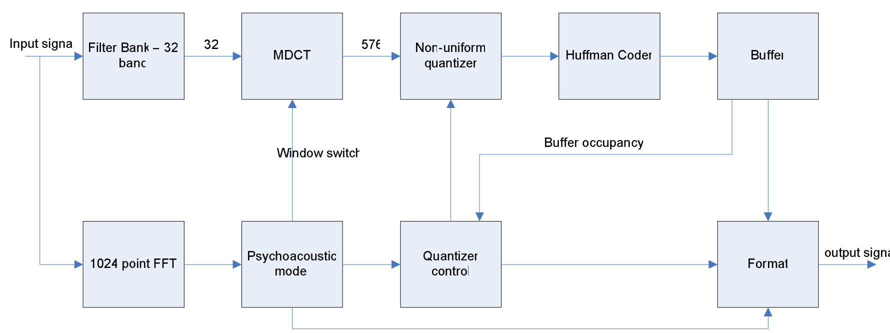

Figure 13: MP3 Encoder

The first difference in Layer 3 is that it employs a Modified Discrete Cosine Transform (MDCT) filterbank, which is based on a 50% overlap between successive blocks. This filterbank has a higher transform coding gain and corresponds better to the bandpass response of the human ear [20].

Non-uniform quantizing is used, where the quantizing step increases as the coefficients increase, which corresponds to critical band analysis. The quantized coefficients are than further compressed using the Huffman coding lossless technique. Layer 3 also uses a buffer memory so that pre-echo can be avoided, and allows different frames to contain different amounts of data.

MPEG-2 AAC

The IOC/IEC MPEG-2 AAC audio coding standard delivers better audio quality than the MPEG-1 family of encoders at below 64 kbps/channel. The block diagram for the MPEG-2 encoder is shown below in Figure 14.

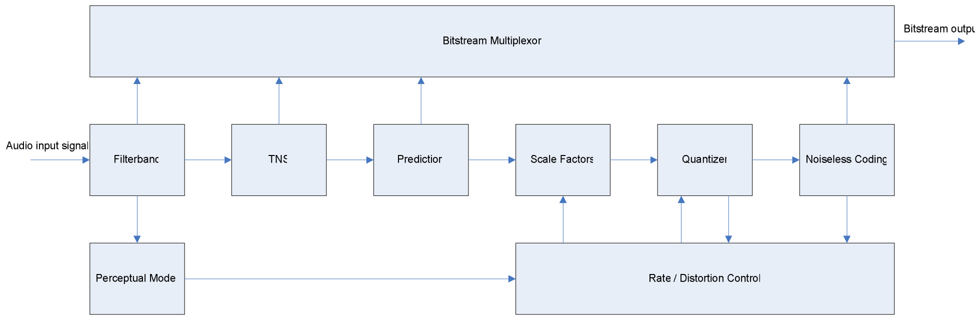

Figure 14: AAC Encoder

The advantages of AAC is that it supports multiple audio channels including subwoofer channels and embedded data channel.. It delivers superior audio quality to that of MPEG-1 family by combining an improved filter bank, backward-adaptive prediction, joint stereo coding, Huffman coding, and supports application specific functionality [21].

Just like the MPEG-1 family, it starts out by sending the audio signal through a filter bank which breaks the signal into subbands. Unlike the other standards, the filter bank uses resolution switching to switch between high-frequency-resolution mode (1024) and high-time-resolution mode of 128 bands [22]. The temporal noise shaping tool (TNS) modifies the filter bank characteristics so that it is better able to adapt to the time-frequency characteristics of the signal. The signal is than analyzed using the perceptual model, which is similar to the psychoacoustic model of the MPEG-1 family. In additional to analysis from the psychoacoustic model, another compression scheme called backward adaptive prediction is used to remove additional redundancies. Quantization and noiseless coding works together to quantize the spectral components of the signal and to remove and additional redundancies in the vector coefficients using Huffman Coding. These two tools work together to achieve optimum quantization while controlling the acceptable noise level. Finally, the rate distortion control adjusts the scale factors to allow more noise into the quantized signal if necessary, further reducing the number of bits required to encode each audio frame.

Dolby AC-3

The Dolby AC-3, developed by Dolby Laboratories, is the coder selected in the United States high definition television (HDTV) standard and widely adopted as the audio codec for DVD films [23]. A block diagram of the Dolby AC-3 encoder is shown below in Figure 15.

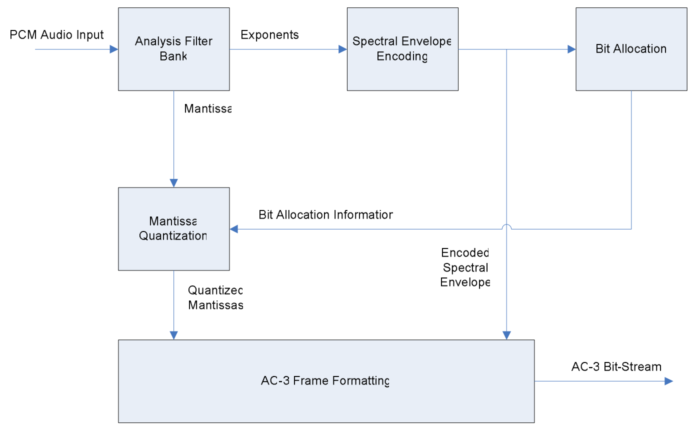

Figure 15: Dolby AC-3 Encoder

The Dolby AC-3 scheme initially transforms the audio sequence into a domain referred to as spectral domain. Each spectral line is represented as a floating point number which consists of an exponent and mantissa. Exponents are coded using a psychoacoustic model which calculated the perceptual resolution. Finally, the information of the perceptual resolution and available bits are used to decide the appropriate quantization manner to quantize the mantissa.

Exponent coding strategies available in AC-3 are referred to as D15, D25, D45, and REUSE. The coding strategy D15 uses the finest frequency resolution and therefore requires the largest number of bits. D45 uses the coarsest frequency and requires the largest amount of bits. The REUSE strategy indicates that the exponents of the current block are the same as the previous block.

The Dolby AC-3 compression scheme makes use of a Modified Discrete Cosine Transform (MDCT) to break the signal up into a series of coefficients. The set of exponents are encoded into a representation of the signal spectrum referred to as the spectral envelope. This envelope is used by the bit allocation scheme to determine how many bits should be used to represent each exponent. The spectral envelope and the coarsely grained mantissa for 6 audio blocks (1536 samples per channel) is formatted into an AC-3 frame.

Sony ATRAC

The ATRAC (Adaptive Transform Acoustic Coder) was developed by Sony and is used in their popular MiniDisc device. ATRAC uses a combination of sub-band coding and modified cosine transform (MDCT). Figure 16 shows the block diagram for the ATRAC encoder.

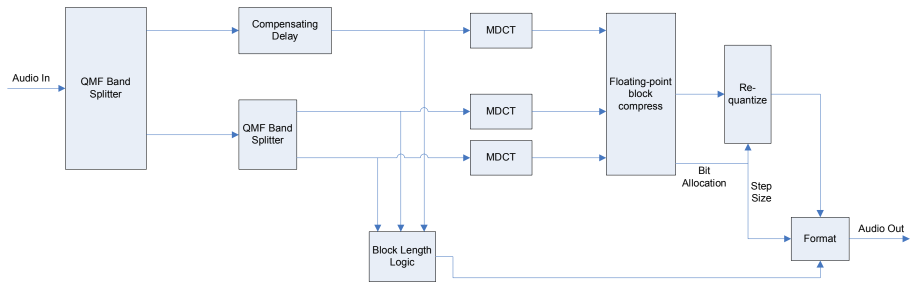

Figure 16: Sony ATRAC Encoder

Similar to MPEG audio coders, the bit stream passes through a filterbank, or a QMF Band Splitter, which splits the spectrum in half. The lower half is than split once more, and the upper half goes through a compensating delay. Each frequency band is formed into blocks, which than pass through a modified discrete cosine transform. The frequencies are group into a total of 52 frequency bands and assigned varying bandwidths based on the width of the critical bands. The coefficients in each frequency bin are than re-quantized on a bit-allocation basis using a masking model.

7. DIGITAL RIGHTS MANAGEMENT

7.1. Need for Digital Rights Management

Compression has brought about new opportunities for digital audio, such as the ability to purchase and download digital content over the internet as opposed to its traditional model of purchasing music on storage mediums such as the CD. As attractive as this option may be, there are major obstacles for digital audio distribution such as the ability for unlimited consecutive of copying, which threatens intellectual property rights. The advent of rogue peer-to-peer music exchange services like Napster [24] has demonstrated the detrimental effects on the music industry when digital content is allowed to circulate through the internet without mechanisms to protect the intellectual property rights. In order for the purchasing of music over the internet to be effective, digital rights management techniques are needed for content protection. These systems need to incorporate aspects of encryption, conditional access, copy control, and media identification and tracing [25].

7.2. Digital Rights Management Technologies

Cryptography-based Protection

Cryptography-based protection includes both an encryption algorithm and key management systems [26]. Protected content is usually encrypted using a symmetric-key algorithm, such as the Advanced Encryption Standard (AES). Many DRM systems use this technique effectively by maintaining the secrecy of the algorithm used to encrypt the content. Although this “security through obscurity” directly goes against Kerckhoffs Principle [27][28], which states that a reverse engineering approach should not be able to break the encryption, many companies have successfully used this solution. Asymmetric encryption uses a series of keys to encrypt the content. These keys are held by both the transmitter and receiver at least one which is computationally infeasible to derive from the other. The decryption key can either come from a secure environment inside the media player, or in the form of a digital license from a licensing server.

Digital Licenses for E-commerce Systems

Fully integrated digital rights management systems allow for both content protection and an e-commerce system. Content is usually protected through a digital license containing content decryption keys and usage rules. These usage rules allow the distributor to specify how the content should be distributed to the user, such as a pay-per-view, one-week rental, etc. These systems include a clearinghouse which handles financial payments, content providers who license the digital content, and a distributor to facilitate this process through an online shop or website. Figure 17 shows a typical e-commerce system that uses digital licensing [28].

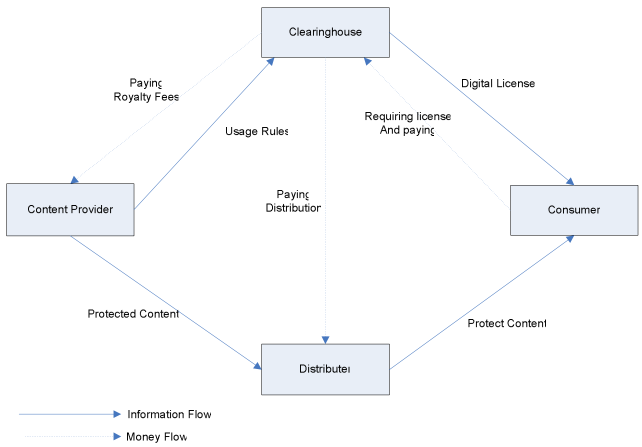

Figure 17: E-commerce System using Digital Licensing

The *content provider* such as a music record label the digital rights of the content and wants to protect these rights.

The *distributor* provides distribution channels, such as an online shop. The distributor receives the digital content from the content provider and creates a web catalogue

The *consumer* uses the system to consume the digital content by retrieving downloadable or streaming content through the distribution channel and than paying for the license through the clearinghouse.

The *clearinghouse* handles the financial transaction for issuing the digital license to the consumer and pays royalty fees to the content provider and distribution fees to the distributor accordingly. .

Digital Watermarking

Digital watermarking is a technique that enables the embedding of hidden data in digital content [28][29]. This can be used for copy protection, or binding information to digital content, such as the buyer to the content. The basic requirements that a watermarking technology must follow are:
Imperceptibility – the watermark must not impair the quality of the audio signal.
Security – The watermark should only be accessible to authorized parties.
Robustness – The watermark must stand up to malicious attacks which have the intention of removing the security and gaining unrestricted access to the content.

The basic idea of watermarking is to apply very slight changes to the individual entities (samples, in the case of audio) in order to ensure imperceptibility. Watermark information is spread in tiny amounts through out the content to make it hard to remove. In order to prevent the watermark from being accessible to unauthorized parties, a secure cryptographic key is typically used. There are two different types of watermarks, robust watermarks and fragile watermarks. Robust watermarks allow the content to be recovered even if the data has been considerably modified. Fragile watermarking can be used as a seal to determine if the data has been modified, and how it has been modified.

Digital watermarking uses a key that is shared by both the encoder and decoder. To create the watermark, the needed parameters are the watermark, original content, and key. Putting this information into the decoder produces the marked data as shown in Figure 18 [30].

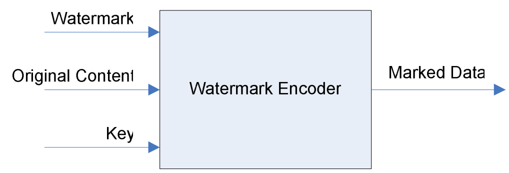

Figure 18 : Generic Watermark Encoder

One powerful application of watermarking is copy protection of content. The goal of copy-control is to prevent uncontrollable and unrestricted copying of the content to other media devices. For this purpose a copy bit is implemented which is not removable and tied to the content. The copy bit is implemented as a robust watermark. In the case of DVD, bits are used to indicate copy-free (CF), copy-never (CN), copy-once (CO), and copy-no-more (CM) [31]. The copy protection system must be allowed to change the state of the bits while not allowing unauthorized parties to manipulate or remove these bits. Another requirement, as with all watermarking technology, is that the copy bits do not perceptually degrade the content.

7.3. Industry Standards

Secure Digital Music Initiative (SDMI)

The Secure Digital Music Initiative is an international consortium whose aim is to develop an open standard for playing, storing, and distributing digital music. The SDMI specification reflects the needs of the music to define a system that prevents redistribution of content over the internet. The initial focus of the project was on portable hardware devices, as demonstrated in the SDMI specification Portable Device Specification Part I, Version 1 [32].

A large part of the SDMI standard relies on robust and fragile digital watermarks for content protection [33]. The standard defines that digital watermarking from ARIS Technologies [34] will be used. The original released music contains both a fragile and roust watermark. During the compression phase of the music, the fragile watermark vanishes and the SDMI-compliant devices use an encryption scheme to bind the content to the device of the legitimate user. PC's and other devices accept content that has both watermarks but refuse to play content that is missing the fragile watermark, indicating that the content is an illegal copy.

In September 2000, SDMI announced a three week public challenge called HackSDMI, inviting the public to evaluate and attempt to break four watermarking techniques being considered for it's copy protection system [35]. Researchers from Princeton University, Rice University, and Xerox Labs participated in the challenge and were able to demonstrate significant progress in leading successful attacks on the digital content. The challenge pointed out weaknesses in watermarking designs to effective attacks, and that using large amounts of information regarding the embedding mechanism of the watermark, extremely powerful attacks can be carried out. While watermarking may be a viable solution in non-hostile or semi-hostile environment, the HackSDMI challenge demonstrates the ease at which even complex watermarking technologies can be circumvented. These reasons have led to this specification being largely abandoned by the industry.

MPEG-4 (ISO/IEC 14496) Intellectual Property Management and Protection
The MPEG-4 standard recognizes the need for protection of content. The Intellectural Property Management and Protection incorporates two distinct pieces: one for the identification of the copyright, and one to enable the protection of the content [25][36].

The identification piece identifies whether the content is protected by a DRM system, what the type of content is, and other information important in identifying the content. The MPEG-4 standard does not enforce when and how often to use these descriptors, or whether this information is present or correct at all.

The second piece is the protection part. Discussion among the MPEG group concluded that it was not feasible to define a full DRM system within the standard because of the great diversity in applications, differing needs for DRM systems, and the legal implications of recommending a certain technology that could later be proved inadequate. Instead, a series of "hooks" are tightly integrated into the MPEG-4 Systems layer. This allows third-party groups to use application specific digital rights management technologies at these various points. The IPMP interface can be extended by other applications through an IPMP toolset. This interface allows for personalized IPMP schemes to be developed and implements at various control points, shown below in Figure 19.

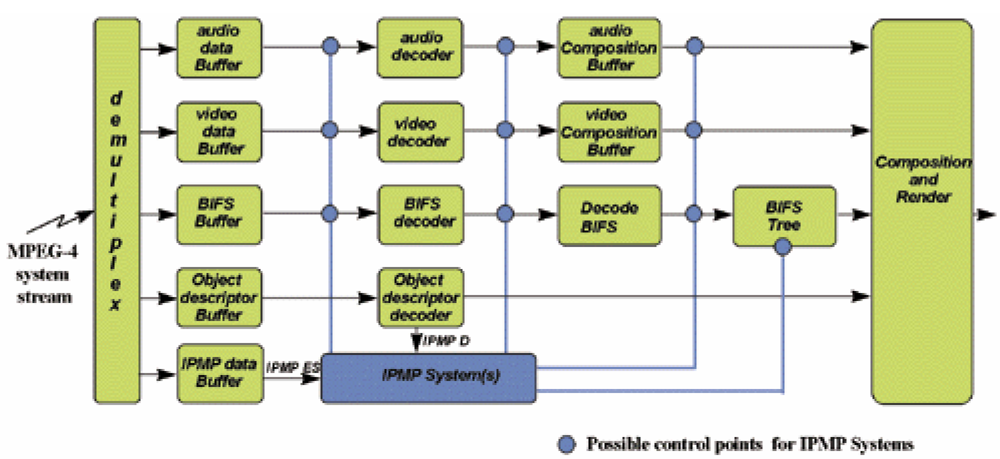

Figure 19: MPEG-4 IPMP

After the standard was released, many voiced concerns that because different digital rights management techniques could be implemented by different manufacturers, that many of them would not work together. This has already begun to occur, with popular portable music players implementing their own DRM schemes that only work with their specific devices. Although interpretability was one of the major reasons for standardizing an IPMP architecture, this has not been the case.

8. CONCLUSION

This paper has discussed current techniques that are being used to provide audio compression in various formats. Most audio coders are based on the general design of a perceptual coder, where properties of the human hearing system are analyzed and perceptually inaudible information is removed. These coders are usually combined with lossless techniques, which remove redundancy, to provide a high compression ratio at audio quality that is close to the original source signal.

There are challenges that lay ahead for compression technologies. Perceptual audio coders, while currently able to provide sufficient compression by removing perceptually inaudible information, will continue to be pushed to allow for higher levels of compression at better quality. When the limit is reached as to how much information can be removed without significant loss of audio quality, new methods beyond that of traditional perceptual audio coders will need to be explored.

Today’s compression technologies are also being asked to provide sufficient content protection so that music can continue to be transferred through mediums such as the internet while preserving the intellectual property rights. This is crucial for innovation in the music industry to continue, as people are motivated to innovate only when they know that the intellectual property rights of their product or content can be sufficiently honored. Digital rights management techniques so far have proved either moderately successful or not successful at all. More research and development needs to take place in the digital rights management arena to ensure that content can be protected through the internet.

9. ACKNOWLEDGEMENTS

I would like to thank Dr. Chance Glenn for agreeing to sponsor me for this independent investigation on digital audio compression and digital rights management. It has been an excellent addition to my curriculum to be able to explore this topic in detail and to be able to apply some of the principles that I have learned to the research and cooperative education opportunities that I have done with him.

I would also like to thank the Department of Computer Engineering at RIT for allowing me to pursue this independent investigation outside of my department. It has added a lot of value to my academic studies to be able interact and learn from professors both in the Computer Engineering Department and outside in various colleges at RIT.

10. REFERENCES

[1]. |Online| Available at http://history.acusd.edu/gen/recording/notes.html

[2]. John Watkinson, *The Art of Digital Audio: Third Addition* (Oxford, UK: Focal Press, 2001), pg. 196-272.

[3]. Ken C. Pohlmann, *Principles of Digital Audio: Third Edition*, (New York, NY: McGraw-Hill, 2000)

[4]. John Strawn, *Digital Audio Engineering: An Anthology*, (Los Altos, CA: William Kaufmann Inc., 1985), pg. 1-30.

[5]. Francis Rumsey, *Digital Audio Operations* (Oxford, UK: Focal Press, 1991)

[6]. |Online| Available at http://www.borg.com/~jglatt/tech/wave.htm

[7]. |Online| Available at http://ccrma.stanford.edu/courses/422/projects/WaveFormat

[8]. W. Kinser. "Compression and It's Metrics for Multimedia", *Proceedings of the First IEEE International Conference on Cognitive Informatics (ICCE'02).*

[9]. Jerry D. Gibson, Toby Berger, Tom Lookabaugh, Dave Lindbergh, Richard L. Baker, *Digital Compression for Multimedia: Principles and Standards* (San Francisco, CA: Morgan Kaufmann Publishers, Inc, 1998)

[10]. Mat Hans, Ronald W. Shafer, "Lossless Compression of Digital Audio", *IEEE Signal Processing Magazine, July 2001.*

[11]. Kalid Sayood, *Introduction to Data Compression* (San Diego, CA: Academic Press, 1996).

[12]. Reza Hashemian, "Condensed Table of Huffman Coding, a New Approach to Efficient Decoding", *IEEE Transactions on Communications, Vol. 52, No. 1, January 2004.*

[13]. Abraham Bookstein, Shmuel T. Klein, Timo Raita, "Is Huffman Coding Dead?", *Annual ACM Conference on Research and Development in Information Retrieval. Proceedings of the 16th International ACM SIGIR conference on Research and development in information retrieval, 1993.*

[14]. Ian H. Witten, Radford M. Neal, John G. Cleary. "Arithmetic Coding for Data Compression", *Communications of the ACM, Volume 30, Issue 6, June 1987.* pg 520-540.

[15]. Nasir Memon, Xuan Kong, Judit Cinkler, "Content-Based Lossless and Near-Lossless Compression of EEG Signals". *IEEE Transactions on Information Technology in BioMedicine, Vol. 3, Num. 3, September 1999.*

[16]. Davis Yen Pan, "Digital Audio Compression, *Digital Technical Journal Vol. 5 Num. 2, Spring 1993.*

[17]. E. Zwicker and R. Feldtkeller, Das Ohr als Nachrichtenempfanger. Stuttgart: S. Hirzel Verlag, 1967.

[18]. |Online| Available at: http://www.umiacs.umd.edu/~desin/Speech1/node10.html

[19]. |Online| Available at: http://www.rarewares.org/wiki/index.php/MPEG1_Layer_3

[20]. Peter. Noll, "MPEG Digital Audio Coding", *Signal Processing Magazine, IEEE, Volume: 14, Issue:5, Sept. 1997.*

[21]. Atul Puri, Tsuhan Chen. *Multimedia Systems, Standards, and Networks.* (Marcel Dekker Incorporated, 2000).

[22]. Karlheinz Bandenburg, "MP3 and AAC explained", *Proceedings of AES 17th International Conference on High Quality Audio Coding, Florence, September 1999.*

[23]. Chi-Min Liu, Szu-Wei Lee and Wen-Chieh Lee. "Bit Allocation Methods for AC-3 Encoder", *IEEE Transactions on Consumer Electronics, Vol. 44, No. 3.* August 1998.

[24]. |Online| Available at: http://www.napster.com

[25]. Frank Hartung and Friedhelm Ramme. "Digital Rights Management and Watermarking of Multimedia Content for M-Commerce Applications", *IEEE Communications Magazine,* November 2000.

[26]. Willem Jonker and Jean-Paul Linnartz. "Digital Rights Management in Consumer Electronics Products", *IEEE Signal Processing Magazine,* March 2004.

[27]. |Online| Available at: http://www.free-definition.com/Kerckhoffs'-Law.html

[28] Qiong Liu, Reihaneh Safavi-Naini, and Nicholas Paul Sheppard. "Digital Rights Management and Content Distribution", *Proceeding of the Austalasian Information Security Workshop Conference on ACSW Frontiers 2003 – Volume 21.*

[29]. A. O. Waller, G. Jones, T. Whitley, J. Edwards, D. Kaleshi, A. Munro, B. MacFarlane, and A. Wood. "Securing the Delivery of Digital Content over the Internet", *Electronics and Communication Engineering Journal, October 2002.*

[30] Michael Arnold, Martin Schmucker, Stephen D. Wolthusen. *Techniques and Applications of Digital Watermarking and Content Protection* (Norwood, MA: Artech House, 2003)

[31] Benoit Macq, Jana Dittmann, Edward J. Delp. "Benchmarking of Image Watermarking Algorithms for Digital Rights Management", *Proceedings of the IEEE, Vol. 92, No. 6, June 2004*.

[32]. |Online| Available at: http://www.sdmi.org.

[33]. Willem Jonker, Jean-Paul Linnartz, "Digital Rights Management in Consumer Electronics Products", *IEEE Signal Processing Magazine, March 2004.*

[34]. |Online| Available at: http://www.aristech.com

[35]. Min Wu, Scott A. Craver, Edward W. Felten, Bede Liu, "Analysis of Attacks on SDMI Audio Watermarks", *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on , Volume: 3 , 7-11 May 2001. pg 1369 - 1372 vol.3*

[36]. |Online| Available at: http://www.w3.org/2000/12/drm-ws/pp/koenen.pdf