voltage can be allowed to decrease and the time it takes to reach zero can be timed with a counter that generates an output digital word. Successive approximation and parallel methods are examples of the former; integration methods are examples of the latter. Oversampling, or decimating A/D converters, are considered in Chapter 5.

## Analog-to-Digital Converter Requirements

The A/D converter must perform a complete conversion at each sample time—for example, 48,000 conversions per second per channel in a professional audio digitization system. Furthermore, the digital word it provides must be an accurate representation of the input binary voltage. In a 16-bit linear PCM system, each of the 65,536 intervals must be evenly spaced throughout the amplitude range, so that even the least significant bits in the resulting word are meaningful. Thus, speed and accuracy are key requirements for any A/D converter. Of course, any A/D converter will have an error of ±½ LSB—an inherent limitation of the quantization process itself.

The time it takes for an A/D converter to output each digital word is called its conversion time. For an A/D converter used in an audio digitization system, conversion time must be within the span of one sampling period. It is sometimes difficult to achieve accurate conversion from sample to sample because of settling time or propagation time errors. The result of accomplishing one conversion may influence the next. If a converter's input moves from voltage A to B and then later from C to B, the resulting digital output for B may be different because of the device's inability to properly settle in preparation for the next measurement. Obviously, dynamic errors grow more severe with demand for higher conversion speed. In practice, speeds required for full fidelity audio digitization can be achieved. Indeed, some A/D converters simultaneously process two waveforms, alternating between left and right channels; however, cost is always relatively high for any A/D with fast conversion time.

Accuracy is another important consideration. Several specifications have been devised to evaluate the performance of A/D converters. Integral linearity measures the "straighiness" of an A/D converter's output. It describes how close the transition voltage points—the analog input voltages at which the digital output changes from one code to the next—are to a straight line drawn through them. In other words, linearity specifies the deviation of an actual bit transition from the ideal transition value, at any level over the range of the converter. Integral linearity is illustrated in figure 4-20. Linearity is tested and the reference line is drawn across the converter's full output range. Integral linearity is the most important A/D specification and is not adjustable. An n-bit converter is not a true n-bit converter unless it guarantees at least ±½ LSB linearity. The converter in figure 4-20 has a ±¼ LSB integral linearity.

Differential linearity error is a measure of the distance between transition voltages—that is, the widths of input voltage bands. Differential linearity is illustrated in figure 4-21. Ideally, all of the bands of an A/D transfer function should be 1 LSB wide. A maximum differential linearity error of ±½ LSB means that the input voltage may have to increase or decrease as little as ½ LSB or as much as 1½ LSB before an output transition occurs. If this specification were exceeded, to perhaps ±1 LSB, some levels could be 2 LSBs wide and others would be 0 LSB wide; in other words, that output code would not exist. The converter in figure 4-21 has an error of ±¾ LSB; some levels are +¼ LSB wide, others are +1¼ LSB wide. Conversion speed can affect both linearity and differential linearity errors.

**Fig. 4-20.** Integral linearity specification of an A/D converter

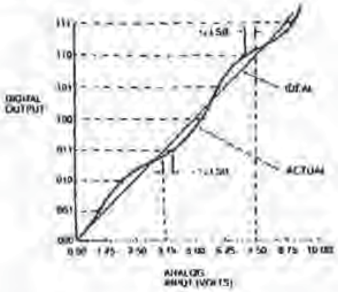
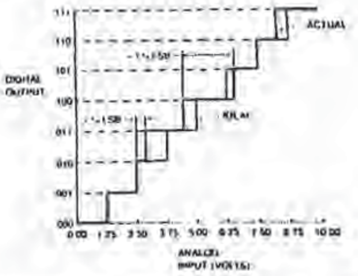


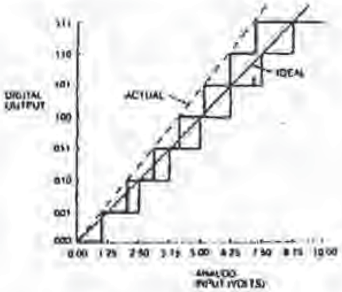
**Fig. 4-21.** Differential linearity specification of an A/D converter



Absolute accuracy error, shown in figure 4-22, is the difference between the supposed level at which a digital transition occurs and where it actually occurs. A good A/D converter should have an error of less than ±½ LSB. Offset voltage, gain error, or noise error can affect this specification. For the converter in figure 4-22, each interval is +¼ LSB in error. In practice, otherwise good A/D devices can sometimes drift with temperature variations and thus introduce inaccuracies.

Code width, sometimes called quantum, is defined as the range of analog input values for which a given output code will occur. The ideal code width is 1 LSB. A/D converters can exhibit an offset error as well as gain error. An A/D connected for unipolar operation has an analog input range from 0 volts to positive full scale. The first output code transition should occur at an analog

Fig. 4-22. Absolute accuracy specification of an A/D converter



input value of ½ LSB above 0 volts. Unipolar offset error is defined as the deviation of the actual transition value from the ideal value. When connected in a bipolar configuration, bipolar offset is set at the first transition value above the negative full scale value. Bipolar offset error is the deviation of the actual transition value from the ideal transition value at ½ LSB above the negative full scale value. Gain error is the deviation of the actual analog value at the last transition point from the ideal value, where the last output code transition occurs for an analog input value 1½ LSB below the nominal positive full scale value. Gain and offset errors are often trimmed at the factory, and may be further zeroed using external potentiometers. Multiturn potentiometers are recommended for minimum drift over temperature and time.

The analog input signal should be scaled to be as close to the maximum input range as possible, to utilize the converter's maximum signal resolution. Generally, a converter uses a low input impedance, which should be driven by a very low impedance (e.g., output of a wide-band, fast-settling operational amplifier) source. Transitions in an A/D converter's input current may be caused by changes in the output current of the internal D/A converter as it tests bits. The output voltage of the driving source must remain constant while supplying these fast current changes.

Changes in the DC power supply will affect an A/D converter's accuracy. Power supply deviations can cause changes in the positive full scale value. This change results in a proportional change in all code transition values—that is, a gain error. Normally, regulated power supplies with 1% or less ripple are recommended. Power supplies should be bypassed with a capacitor (e.g., 1 to 10 microfarad tantalum) located close to the converter, to obtain noise-free operation. Noise and spikes from a switching power supply must be carefully filtered.

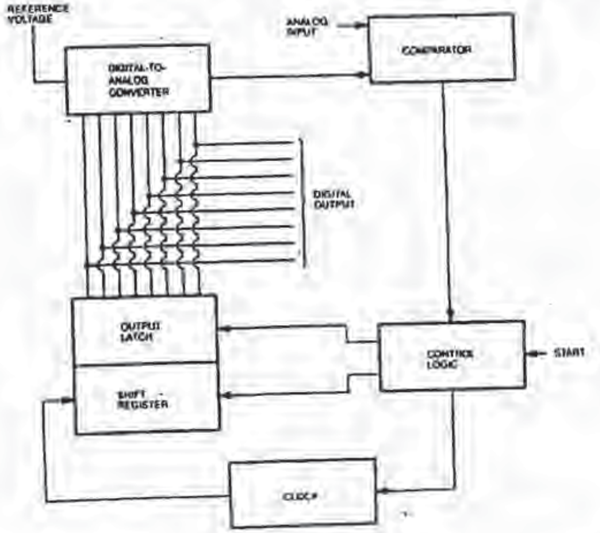
Quality A/D converters are guaranteed to be monotonic; that is, the output code either increases or remains the same for increasing analog input signals. In addition, good A/D converters are assured of having no missing codes over a specified temperature range.

## *Successive Approximation Analog-to-Digital Converter*

There are many types of A/D circuit design appropriate for various applications. For audio digitization, the necessity for both speed and accuracy limits the choices to a few types. The classic A/D converter used in audio digitization is the successive approximation register (SAR) A/D design. It is shown in the block diagram in figure 4-23. This converter employs a digital-to-analog converter in a feedback loop, a comparator, and a control section. In essence, this converter compares the analog input with its interim digital word converted into analog, until the two agree within the given resolution. In operation, the device follows an algorithm that, bit by bit, sets the output digital word to match the analog input.

For example, let's assume an analog input of 6.92 volts and an 8-bit A/D converter. The operational steps of the SAR converter are shown in figure 4-24. The most significant bit in the SAR is set to 1, with the other bits still at 0; thus the word 10000000 is applied to the internal D/A. This word places the D/A output at its half value of 5 volts. Since the input analog voltage is greater than the D/A output, the comparator remains high. The first bit is stored at logical 1. The next most significant bit is set to 1 and the word 11000000 is applied to the D/A, with an interim output of 7.5 volts. This is too high, so the second bit is reset to 0 and stored. The third bit is set to 1, and the word 10100000 is applied to the D/A; this produces 6.25 volts, so

Fig. 4-23. Block diagram of a successive approximation register A/D converter



### Format

The transformation process from raw data to coded data is dependent on the selection of format. Each of these record processing steps may be carried out in many ways, and the way in which the resulting data is assembled may be variously determined. A digital recording may consist of many frames of data, arbitrarily interleaved. Each frame consists of group codes, such as synchronization, address, identification, data, and redundancy. Each frame contains many data words, including samples that contain the time-multiplexed bits of audio data. There is obviously considerable latitude involved in determination of a format, and the relative efficiency and success of each are not equal. Additionally, the choice of medium strongly influences format design. This is discussed in further detail in following chapters.

### Record Modulation

Record modulation processing is the final electronic manipulation of the audio data before its storage. Because digital audio is commonly considered to involve the storage of 1s and 0s, it may be surprising to learn that the binary code is not recorded directly. Rather, a modulated code is stored, which represents the bit stream. It is thus a modulation waveform that is recorded and interpreted upon playback to recover the original binary data and thus the audio waveform. Modulation facilitates data reading by further delineating the recorded logical states. Moreover, through modulation, a higher coding efficiency is achieved; although more bits may be recorded, a greater data density can be achieved overall. On the other hand, different modulation codes precipitate incompatibility among digital recording media.

### Recording

Following modulation, the data is ready for storage on the medium. In the case of a stationary head digital recorder, the data is applied to a recording circuit, which generates the current necessary for saturation recording. The flux reversals recorded on the tape thus represent the bit transitions of the modulated data. The recorded waveforms may appear highly distorted; this does not affect the integrity of the data, and permits the recording of higher densities. In optical systems such as the compact disc, a previously recorded digital tape is played through a laser cutting machine, which produces the master glass plate used in CD manufacturing. The modulation code results in pits. Each pit edge represents a binary 1, while spaces between represent binary 0s. In any event, storage to media or other real-time digital audio processing marks the end of the digital recording chain.

# 5 Digital Audio Reproduction

In an audio digitization system, the recording and reproduction processors serve as input and output transducers, converting the analog audio waveform into a signal suitable for digital storage, then reconverting the stored or processed signal to analog form. In a linear pulse code modulation (PCM) digitization system, the functions of subsystems on the reproduction side of the signal path are largely reversed from those on the record side. The reproduction subsystems include the demodulation circuit, reproduction processing circuits, demultiplexer, digital-to-analog converter, output sample and hold circuit, and output low-pass filter. This chapter describes the reproduction circuits used in a linear PCM audio digitization system, as shown in figure 5-1. Oversampling techniques are considered as well.

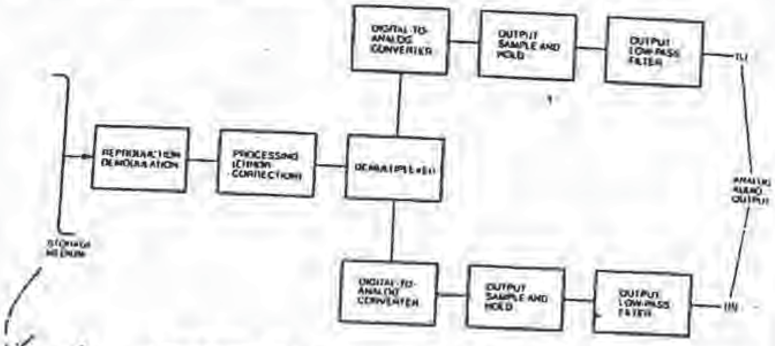
## Demodulation Circuits

The demodulation circuits are the first step in reproduction of the digital audio signal in which the coded waveform recorded on the medium is again converted to an analog signal. The demodulation circuits must accomplish several important functions. The signal derived from the medium is of very low amplitude and must be amplified. This waveform is highly distorted and must be processed to recover the data. Finally, the data must be synchronized and demodulated to restore the original literal data.

### Operation of Demodulation Circuits

A preamplifier is required to boost the signal from the medium. The signal is so low in level that processing can be accomplished only after amplification. To achieve high recording density, the fidelity of the modulation code wave-

Fig. 5-1. Linear PCM reproduction section



hardware
chip
DRAM (clock)
DSP (?)
storage space
(n-way)

form as recorded on the medium has been allowed to deteriorate. Thus, the signal from the medium does not have the clean characteristics of the original data. Rather, the amplitudes of the recorded data as read from the tape head are rounded, and only the transitions between levels correspond to the original signal. A waveform shaper circuit is used to identify the transitions and reconstruct the 1s and 0s of the signals. In this way, data can be entirely recovered with no penalty for the waveform's deterioration. The data is again as clean as if it had been literally recorded, but storage of a much greater amount of data has been permitted.

The music signal data and its error correction code are identified and separated from the peripheral data, which is additionally identified and separated into frame synchronization and bit synchronization signals. Frame synchronization pulses are used to identify individual frames. Bit synchronization pulses are derived and used to identify individual bits within each frame and to synchronize the playback signal, thus determining the 1 or 0 content of each pulse. The modulated music signal data—whether it is HDM-1, EFM, MFM, or another code—is typically demodulated to NRZ code, that is, a simple code in which amplitude level represents the binary information. The method for interpreting NRZ is to read a logical 1 when there is a high amplitude and logical 0 when there is a low level amplitude. The music data has thus regained its original binary form and is ready for further reproduction processing. Modulation codes are discussed further in Chapter 7.

## Reproduction Processing

The reproduction processing circuits are primarily concerned with minimizing the effects of data storage. Every storage medium suffers from limitations, such as mechanical variations and potential for damage to data. With analog storage, the problem must generally be corrected within the medium itself; for example, to minimize wow and flutter, the turntable's speed must be kept accurate. With digital systems, because of the density of the storage, the potential for error as a result of storage is much greater. However, digital encoding also presents the opportunity to correct for many errors. The reproduction processing circuits buffer the data to minimize the effects of mechanical variations in the medium, and to perform error correction. In addition, demultiplexing is performed to restore parallel structure to the audio data.

### *Necessity for Reproduction Processing*

The reproduction processing circuits must accomplish the reverse of the signal manipulations performed on the record side of the digitization chain. In addition (and the primary reason for performing the record processing), the reproduction circuits must check for errors that have occurred during storage. Because of the packing density used in digital recording, errors occur with certainty; only their frequency and severity vary. If within tolerance, errors may be detected and corrected with absolute fidelity to the original data, making digital recording a highly reliable technique. If the nature of the errors exceeds the error correction circuit's ability to correct, estimated values may be substituted for missing or bad data.

Transport problems include rapid variation in timing, low frequency drifting of timing, errors from tape stretching, and improper head alignment. A badly designed digital recorder could additionally cause crosstalk errors in the heads or associated wiring, or errors due to power supplies with electrical noise. Additionally, errors are caused by both manual and electronic tape editing; reproduction processing circuits must be able to accommodate the large disruptions of data that occur at an edit point.

### *Description of Reproduction Processing Circuits*

The reproduction processing circuits must initially de-interleave the data. Prior to recording, the data has been scattered in the bit stream to ensure that a defect in the medium does not affect consecutive data. With de-interleaving, the data is again properly assembled, and errors caused by medium defects are now scattered through the bit stream, where they are easier to correct because of their isolation. The entire interleave and de-interleave process is shown in figure 5-2.

Mechanical instability in the transport will introduce timing errors, such as jitter, as data is read from the medium; this is shown in figure 5-3. To overcome this problem, a data buffer is used. A buffer may be thought of as a pail of water; water is poured into the pail carelessly, but a spigot at the bottom of the pail supplies a constant stream. Specifically, a buffer is a memory into which the data is fed irregularly, as it arrives from the medium. However, the output of the buffer occurs at an accurately controlled rate,

Fig. 5-2. An example of the interleave and de-interleave process

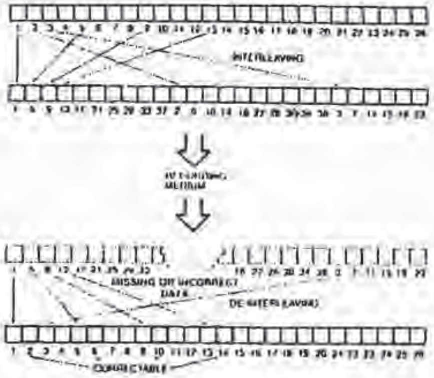
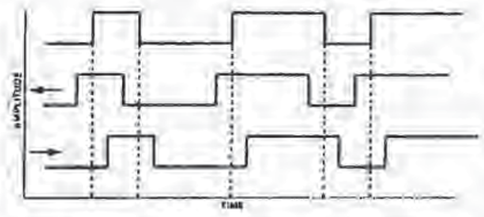


Fig. 5-3. Jitter is the result of any time-axis instability.



ensuring precise data timing. Samples are thus assembled at the same rate at which they were taken, guaranteeing the lossless nature of time sampling.

Using redundancy techniques such as parity and checksums, the data is checked for errors. When the parity bits or checksums calculated do not agree with those read from the medium, an error has occurred either in the audio data or in the parity and checksum data. Several methods are used to isolate the error and determine where the fault has occurred. In the case of bad audio data, error correction techniques are used to recover the correct values. Using parity bits, checksums, or redundant data, the missing values may be determined and substituted. When the error is too extensive for recovery, error compensation techniques are used to conceal the error. Most simply, the last data value can be held until valid data resumes. Linear interpolation is a method of calculating new data to form a bridge over the error. For larger errors, interpolation and other compensation techniques become insufficient, and error concealment becomes marginal; the presumed values differ widely from the lost original values. In extreme cases, when error compensation is not sufficient, the audio signal will be muted until valid data resumes. A more complete discussion of error correction techniques may be found in Chapter 8.

The final circuit in the reproduction processing chain is the demultiplexer. The serial bit stream now consists of the original audio data, or at least as original as the error correction circuitry has achieved. However, one remaining manipulation must be performed on the data to convert it to its parallel form, in which it again appears as discrete words, each representing one sample value. The demultiplexer circuit accepts a serial bit input, counting as the bits are clocked in. When a full word has been received, it outputs all of the bits of the entire word simultaneously, performing its task again and again as the data is applied. An example of a demultiplexer circuit is shown in figure 5-4.

On leaving the reproduction processing circuitry, the data has regained timing stability, been de-interleaved, corrected for errors incurred during storage, and demultiplexed to again form its parallel sample words. The data is now ready for digital-to-analog conversion.

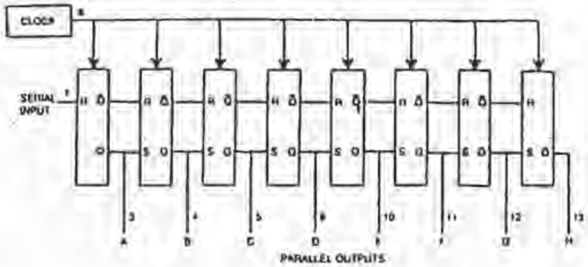
## Digital-to-Analog Conversion

The digital-to-analog (D/A) converter is one of the most critical elements in the reproduction system. Just as the analog-to-digital (A/D) converter largely determines the overall quality of the record system, the digital-to-analog converter determines how accurately the digitized signal will be restored to the analog domain. However, whereas A/D conversion inherently introduces a quantization error, there is no corresponding quantization error in the D/A conversion process. In playback-only systems, such as the compact disc system, the D/A converter must be carefully designed to permit stable operation under varying conditions. Fortunately, several excellently designed D/A converters are available, and those integrated circuits are available at relatively low cost.

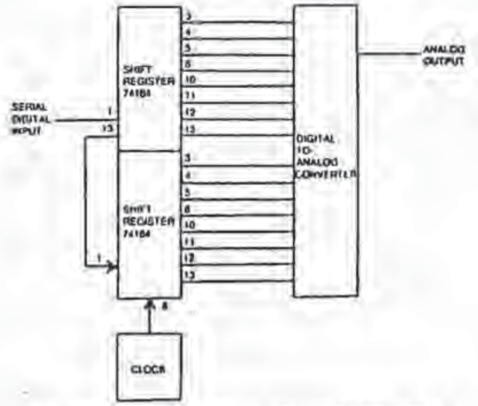
### *Digital-to-Analog Converter Requirements*

The digital-to-analog converter is subject to many of the requirements and is prone to many of the same errors as the analog-to-digital converter, described in Chapter 4. The ideal transfer function, shown in figure 5-5, is more nearly approximated by D/A converters. A D/A converter must exhibit integral linearity; that is, the "straightness" of its transfer from digital to analog must be good. Its differential linearity must maintain an error of less than ± ½ LSB; that is, when the input code changes by one bit, the analog output should change by one voltage step. A D/A converter must be monotonic; that is, the analog output must increase as the digital input increases, and decrease as the input decreases. An example of a D/A converter that is not monotonic is shown in figure 5-6. A D/A converter must have good absolute accuracy, small offset, and fast settling time. The criteria for settling time are shown in figure 5-7. Settling time for a D/A converter is the elapsed time between

**Fig. 5-4.** Parallel output shift registers can be used to convert the serial output of the reproduction circuitry to parallel data prior to D/A conversion.



*(A) Functional block diagram of 74164 shift register*



*(B) Conversion from serial to parallel with cascaded shift registers*

a new input code and the time when the analog output falls within a specified tolerance. In terms of audio performance specifications, these combined requirements constitute a device with low distortion and intermodulation products.

Most importantly, a quality D/A converter must be highly accurate. A 16-bit D/A converter with an output range of ± 10 volts should have a distance between quantization levels of 20/65,536 = 0.00305 volts. For example, the output from the input word 1000000000000000 should be 3 millivolts larger than that from the input word 0111111111111111. In other words, the sum of the lower 15 bits must be accurate to that precision, compared to the value

**Fig. 5-5.** Transfer function for an ideal 3-bit D/A converter

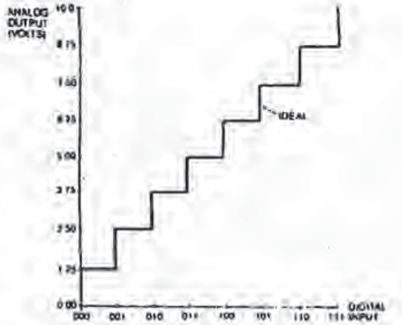


**Fig. 5-6.** Transfer function for a 3-bit D/A converter that is not monotonic

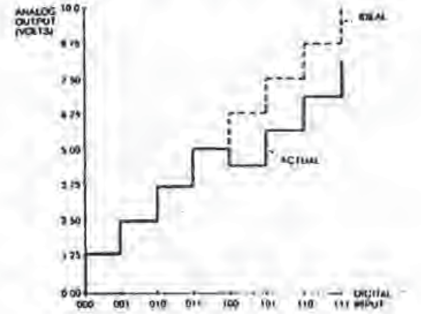


**Fig. 5-7.** The settling time for a D/A converter is measured over the duration of a complete bit change.

of the highest bit. The lower 15 bits should have a relative error of one-half quantization level, or 0.0015%, as should the MSB. Difficulty in achieving accuracy at the center of the D/A converter's range leads to crossover distortion. Moreover, the percentage of error increases when low level audio signals are converted.

Most D/A converters operate with a two's complement input. For example, an 8-bit D/A converter would have a most positive value of 01111111 and a most negative value of 10000000. As we have seen, in this format the MSB is complemented to serve as a sign bit. To accomplish this, the MSB can be inverted before the word is input to the D/A converter, or the D/A converter may have a separate, complementing input for the MSB.

## Weighted Resistor Digital-to-Analog Converter

Many types of digital-to-analog converters are available. Three types are commonly employed in audio digitization systems. To understand their function, we must begin with a simple design that illustrates the operation of the D/A converter. A digital-to-analog converter accepts an input digital word and converts it to an output analog voltage or current. The simplest kind of D/A converter contains a series of resistors and switches, and is known as a weighted resistor D/A converter. An example is shown in figure 5-8.

This type of converter contains a switch for each input bit; the corresponding resistor represents the value associated with that bit. A reference voltage is used to generate currents in the resistors. A digital 1 closes a switch and contributes a current, while a digital 0 causes the switch to remain open and prevents current flow. An operational amplifier sums the currents and converts them to an output voltage. A low value binary word with many 0s keeps many switches open and a small voltage results. A high value word with many 1s closes more switches and a high voltage results. While this design looks good on paper, it is rarely used in practice because of the complexity in manufacturing resistors with sufficient accuracy. Consider this example:

$$V_{out} = -V_{ref}\left(\frac{b1}{2} + \frac{b2}{4} + \frac{b3}{8} + \frac{b4}{16} + \frac{b5}{32} + \frac{b6}{64} + \frac{b7}{128} + \frac{b8}{256}\right)$$

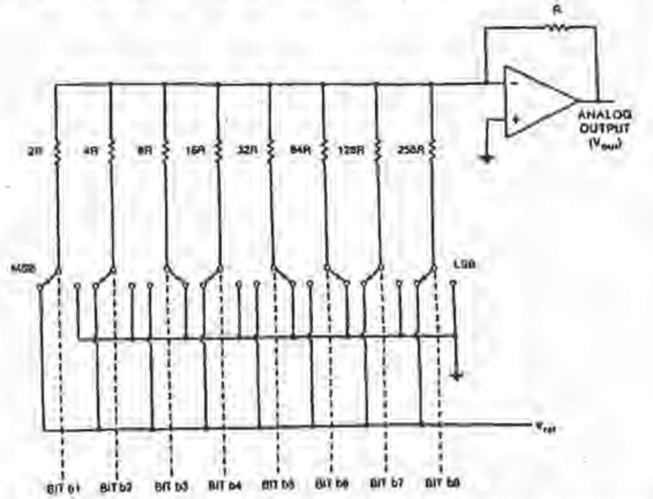
where b1 through b8 represent the input binary bits. For example, suppose the reference voltage is 1, the input word is 11010011, and $V_{ref} = 10v$.

$$V_{out} = -10\left(\frac{1}{2} + \frac{1}{4} + \frac{0}{8} + \frac{1}{16} + \frac{0}{32} + \frac{0}{64} + \frac{1}{128} + \frac{1}{256}\right)$$

$$= -10\left(\frac{1}{2} + \frac{1}{4} + \frac{1}{16} + \frac{1}{128} + \frac{1}{256}\right)$$

$$= -8.24v$$

Fig. 5-8. A weighted resistor D/A converter uses resistors related by powers of two.



Because each next resistor value must be a power of two greater than the previous one, widely varying values result. For example, in a 16-bit D/A converter, the largest-to-smallest resistor ratio is $2^{16} = 65{,}536$. If the smallest resistor value is 1K ohm, the largest is over 65M ohms. Similarly, the smallest current may be 30 nanoamps and the largest 2 milliamps. In short, this design creates conditions that are difficult to meet in manufacturing.

## R-2R Ladder Digital-to-Analog Converter

A more suitable design approach for a D/A converter is the R-2R resistor ladder shown in figure 5-9. This circuit contains resistors and switches; however, there are two resistors per bit. Each switch contributes its appropriately weighted component to the output. The current splits at each node of the ladder, resulting in currents through the switch resistors that are weighted by binary powers of two. If a current I flows from the reference voltage, I/2 flows through the first switch, I/4 through the second switch, I/8 through the third switch, etc. Digital input bits are used to control ladder switches to produce an analog output.

# 13 Digital Signal Processing

In many ways, digital signal processing (DSP) brings us back to the elemental beginning of our discussion of digital audio. Although A/D and D/A conversion, storage, and other topics are critical to any audio digitization system, it is the signal processing of the digital audio data that is germane to the ultimate success of the venture. Without the ability to manipulate the numbers that comprise digital audio data, its digitization and storage would not be useful for most applications. Moreover, a discussion of digital signal processing returns us to the roots of digital audio in that the technology is based on the same logic circuits, gates, adders, and storage cells that first occupied us. On the other hand, digital signal processing is a science quite different from simple data processing, with special algorithms required to achieve its aims of efficient signal manipulation.

## Fundamentals of Digital Signal Processing

A signal can be any natural or artificial phenomenon that varies as a function of some idependent variable. For example, when the variable is time, changes in barometric pressure, temperature, oil pressure, current, or voltage are all signals that can be recorded, transmitted, or manipulated either directly or indirectly. As we observed in Chapter 2, their representation can be either analog or digital in nature. Either representation offers both advantages and disadvantages.

### *Linearity and Time Invariance*

A discrete system is any system that accepts one or more discrete input signals $x(n)$ and produces one or more discrete output signals $y(n)$, in accordance with a set of operating rules. The input and output discrete time signals are represented by a sequence of numbers. If an analog signal x(t) is sampled every T seconds, the discrete time signal is x(nT), where n is an integer. Time can be normalized so that the signal is written as x(n).

Two important criteria for discrete systems are linearity, and time-invariance. Linear systems exhibit the property of superposition; the response of a linear system to a sum of signals is the sum of responses to each individual input. A linear system exhibits the property of homogeneity. The amplitude of the output of a linear system is proportional to that of the input. A linear discrete system with the input signal $ax_1(n) + bx_2(n)$ produces an output signal $ay_1(n) + by_2(n)$, where a and b are constants. The output amplitude is proportional to that of the input, and no new signal components are introduced. As we shall see, all z transforms and Fourier transforms are linear.

A discrete time system is time-invariant if the input signal $x(n - k)$ produces an output signal $y(n - k)$, where k is an integer. In other words, a time-invariant system behaves the same way at all times; for example, an input delayed by k samples generates an output delayed by k samples.

Furthermore, most useful discrete systems are stable. Any arbitrary input signal of finite amplitude produces an output signal of finite amplitude.

A discrete system is causal if at any instant the output signal corresponding to any input signal is independent of the values of the input signal after that instant. In other words, there are no output values before there has been an input signal. As one author puts it, a causal system doesn't laugh until it has been tickled.

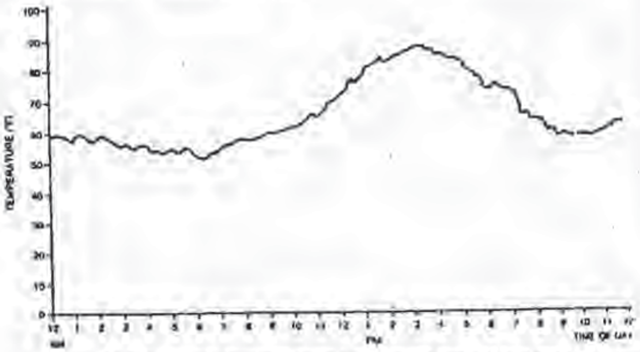
## Digital Representation

When the independent variable, such as time, is continuously variable, the signal is defined at every real value of time (t); the signal is thus a continuous-time signal. For example, figure 13-1(A) shows temperature changes through a 24-hour day. When the signal is only defined at discrete values of time (nT), the signal is a discrete-time signal. Figure 13-1(B) shows the share price of a stock through the trading session, as quoted every hour. In this case, the stock value exists as a continuous variable in time, but we choose to sample it only at certain discrete times. As we observed in Chapter 3, using the sampling theorem, any band-limited continuous-time function can be represented without loss as a discrete-time signal; it is this fact that permits audio digitization. Although general discrete-time signals and digital signals both consist of samples, a general discrete-time signal may take any real value, but a digital signal may only take a finite number of values. With digital audio, in most cases, the measure of signal amplitude entails approximation, or quantization.

## Advantages and Disadvantages

Digital processing of acquired waveforms offers several advantages over processing of continuous-time signals. Fundamentally, the use of unambiguous discrete samples promotes reduced sensitivity to external effects such as

Fig. 13-1. Continuous-time versus discrete-time signals.



(A) The constantly monitored change in temperature over a 24-hour day is an example of a continuous-time signal.

(B) The hourly quotations of a stock price are an example of a discrete time signal.

noise, temperature and aging, use of components with lower tolerances predetermined accuracy, identically reproducable circuits, and a theoretically unlimited number of successive operations on a sample. The programmable nature of discrete-time signals permits changes in function without changes in hardware. Digital integrated circuits are small, highly reliable, low in cost, and capable of complex processing. Some operations implemented with digital processing are difficult or impossible with analog means. Examples include filters with linear phase, long-term uncorrupted memory, adaptive systems, image processing, error correction, and signal transformations. The latter includes time domain to frequency domain transforma tion with the discrete Fourier transform (DFT) and special mathematical processing such as the fast Fourier transform (FFT), to reduce the computing burden.

On the other hand, DSP has some disadvantages. For example, the technology always requires power; there is no passive form of DSP circuitry. Digital signal processing cannot presently be used for very high frequency signals. Digital signal representation of a signal requires a larger bandwidth than the corresponding analog signal. Development of DSP technology is expensive. Circuits capable of performing fast computation are required. Finally, when used for analog applications such as audio, A/D and D/A conversion are required. In addition, the processing of very weak signals such as antenna signals, or very strong signals such as those driving a loudspeaker, presents difficulties; digital signal processing thus requires appropriate amplification treatment of the signal.

## *Applications of DSP*

Some of the earliest uses of digital signal processing included soil analysis in oil and gas exploration, and radio and radar astronomy using mainframe computers. With the advent of specialized hardware, extensive applications in telecommunications were implemented, including modems, data transfer between computers, and vocoders and transmultiplexers in telephony. Medical science has employed digital signal processing in processing of x-ray and NMR images. Image processing is also used for photographs received from orbiting satellites and deep space vehicles. Television studios use digital techniques for manipulating picture signals. Analytical instruments use digital signal transforms such as FFT for spectral and other analysis. The chemical industry uses digital signal processing for industrial process control. Digital signal processing has revolutionized professional audio in terms of effects processing, interfacing, user control, and digital control. The consumer sees digital signal processing in the guise of the compact disc system, digital television receivers, as well as digital radio receivers and telephones. Today, digital signal processing extends through many diverse applications, and in particular is omnipresent in the field of digital audio technology. The list of audio applications on the next page demonstrates that DSP is to be found throughout audio technology.

# Discrete Systems

Digital audio signal processing is concerned with the manipulation of audio samples. Because those samples are digitally represented as numbers, digital audio signal processing is a science of number crunching. Hence, it is inevitable that any fundamental understanding of audio DSP must first tackle its mathematical essence. The good news is that linear algebra and calculus are not required for an understanding of DSP theory. The bad news is that complex numbers are.

**Audio applications for DSP**

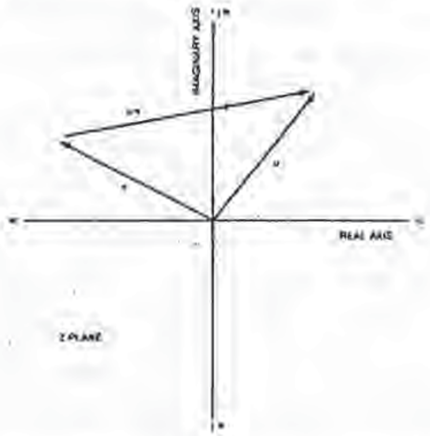
- Transmission and Storage
    - Modulation and demodulation
    - Error correction
    - Error concealment
- System Operation
    - Multiplexing
    - Sample rate conversion
    - Control signals
- Audio Signal Processing
    - Digital filtering
    - Adaptive equalization
    - Reverberation
    - Noise reduction
    - Mixing and editing
    - Acoustic analysis
- Audio Signal Generation
    - Digital audio generators
    - Control of systems

## *Complex Numbers*

Analog and digital networks share a common mathematical basis. Fundamentally, whether the discussion is one of resistors, capacitors, and inductors, or scaling, delay, and addition (all linear, time-invariant elements), processors can be understood through complex numbers. A complex number $z$ is any number that may be written in the form $z = x + jy$, where $x$ and $y$ are real numbers, and where $x$ is the real part and $jy$ is the imaginary part of the complex number. An imaginary number is any real number times $j$, where $j$ is the square root of $-1$. Clearly, there is no number that when multiplied by itself gives a negative number, but it was an easy matter for mathematicians to invent the concept of an imaginary number. (Mathematicians refer to it as $i$ but engineers use $j$, because $i$ denotes current). The form $x + jy$ is the rectangular form of a complex number, and represents the two-dimensional aspects of numbers. For example, the real part can denote distance, and the imaginary part can denote direction. A vector can be easily constructed, clearly showing the indicated location, as shown in figure 13-2.

A waveform could be described by a complex number. (This is often expressed in polar form, with two parameters $r$, $\theta$. The form $re^{j\theta}$ may also be used.) If a dot is placed on a circle and rotated, perhaps representing a waveform changing over time, the dot's location may be expressed by a complex number. A location of 45° would be expressed as $0.707 + 0.707j$. A location of 90° would be $0 + 1j$; 135° would be $-0.707 + 0.707j$; and 180°

Fig. 13-2.
An example of vectors used to describe locations in the complex plane.



would be −1 + 0j. The size of the circle could be used to indicate the magnitude of the number.

Moreover, the j operator can be used to convert between imaginary and real numbers. A real number multiplied by an imaginary number becomes complex, and an imaginary number multiplied by an imaginary number becomes real. Multiplication by a complex number is analagous to phase shifting; for example, multiplication by j represents a 90° phase shift, and multiplication by 0.707 + 0.707j represents a 45° phase shift. In the digital domain, phase shift is performed by time delay. A digital network comprised of delays can be analyzed by changing each delay to a phase shift. For example, a delay of 10° corresponds to the complex number 0.984 − 0.174j. If the input signal were multiplied by this complex number, the output result would be a signal of the same magnitude, but delayed by 10°.

## Complex Plane

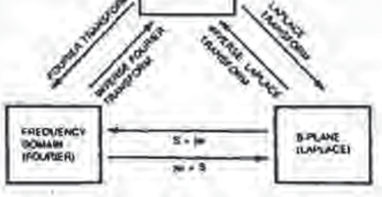
The variable z is complex, and X(z) is the function of the complex variable. The set of z in the complex plane for which the magnitude of X(z) is finite is said to be in the region of convergence. Z is taken to lie on the complex z-plane. The set of z in the complex plane for which the magnitude of X(z) is infinite is said to diverge. The function X(z) is defined over the entire z-plane, but is only valid in the region of convergence. The complex variable s is used to describe complex frequency; this is a function of the Laplace transform. S variables lie on the complex s-plane. The s-plane may be mapped to the z-plane; vertical lines in the s-plane map as circles in the z-plane.

We must design within the region of convergence because we are working with a finite number of samples. The unit circle is the smallest region within the z-plane that falls within the region of convergence for all finite stable sequences. As we shall see later, poles must be placed inside the unit circle of the z-plane for proper stability. Improper placement of the poles may put us in the region of divergence, constituting an instability.

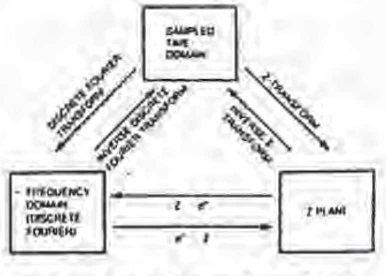
Signal processing, analog or digital—perhaps best exemplified by filter design—can be considered in any of three domains. Together, they offer different perspectives on a unified theory. For analog signals, the three domains are time, frequency, and s-plane. For sampled signals, they are discrete time, discrete frequency, and z-plane. The analog relationships between a continuous signal, its Fourier transform, and Laplace transform are shown in figure 13-3(A). The discrete-time relationships between a discrete signal, its discrete Fourier transform, and z-transform are shown in figure 13-3(B). Transforms are discussed in more detail later in this section.

An important process is mapping from the s- to the z-plane. Theoretically, this function allows the designer to choose an analog transfer function and

Fig. 13-3.
Transforms are used to shift a signal's perspective from one domain to another.



(A) Analog signals can be expressed in time, frequency, and s-plane domains.



(B) Discrete signals can be expressed in sampled time, frequency, and z-plane domains.

find the z-transform of that function. Unfortunately, the s-plane generally does not map into the unit circle of the z-plane; thus, stable analog filters, for example, do not always map into stable digital filters. This is avoided by multiplying by a transform constant, used to match analog and digital frequency response. There is also a nonlinear relationship between analog and digital break frequencies, which must be accounted for. The nonlinear effects are known as warping effects and the use of the constant is known as prewarping the transfer function.

## *Impulse Response and Convolution*

The impulse response is an important concept in many areas, including digital signal processing. The impulse response h(t) gives a full description of a linear system in the time domain. An impulse is considered to be any short duration pulse. When applied to a network such as a filter, an altered impulse, referred to as the network's impulse response, is output. The impulse response reveals the frequency response and phase shift characteristics of the filter. Furthermore, the impulse response can be sampled and used to filter a signal. Audio samples themselves are impulses, represented as numbers. The signal could be filtered, for example, by using the samples as scaling values; all of the values of a filter's impulse response are multiplied by each signal value. This yields a series of filter impulse responses scaled to each signal sample. To obtain the result, each scaled filter impulse response is substituted for its multiplying signal sample. The filter response may extend over many samples; thus, several scaled values may overlap. When these are added together, the series of sums forms the new filtered signal values.

This is the process of convolution. It is a time domain process that is equivalent to the multiplication of the frequency responses of two networks. In short, convolution in the time domain is equivalent to multiplication in the frequency domain. Furthermore, the duality exists such that multiplication in the time domain is equivalent to convolution in the frequency domain. Figure 13-4 shows the correspondence between convolution and multiplication. The effect of filtering a discrete signal can thus be predictably known. A digital filter changes the frequency response of a signal by replacing each signal sample with a scaled replica of the filter impulse response. Similarly, other kinds of digital signal processing can be performed on discrete signals.

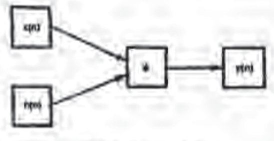
Because convolution is not an intuitive phenomenon, a simpler, graphical illustration of its nature may be useful. Consider the waveform in figure 13-5(A). It can be divided into discrete pieces such that

$$x(t) = x_1(t) + x_2(t) + x_3(t) + \cdots$$

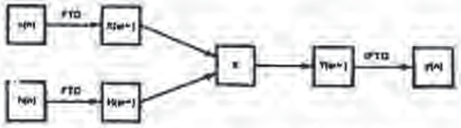
In other words,

$$x(t) = \sum_{k}^{\infty} x_k(t)$$

where k = 1,2,3.



(A) Direct convolution



(B) Multiplication, Fourier transformation, and inverse Fourier transformation

**Fig. 13-4.** Given an input signal x(n) and impulse response h(n) the output signal y(n) may be calculated through (A) direct convolution, or (B) multiplication, Fourier transformation, and inverse Fourier transformation. In practice, the latter method is often an easier calculation.

Consider a network that produces an output h(t) when a single piece of the waveform is input, as shown in figure 13-5(B). The output h(t) defines the network; from this single response we can find the network's response to any input. The network's complete response to the waveform can be found by adding its response to all of the input pieces. The response h(t) to $x_1(t)$ is scaled by the amplitude of $x_1(t)$ and is output time-invariantly with $x_1(t)$. Similarly, the inputs that follow produce outputs that are scaled and delayed by the delay of the input, as shown in figure 13-5(C). The sum of the individual responses is the full response to the input waveform:

$$y(n) = \sum_{k=-\infty}^{\infty} h(k) \cdot x(n-k).$$

This is convolution, mathematically expressed as:

$$y(n) = h(n) * x(n)$$

where * denotes convolution.

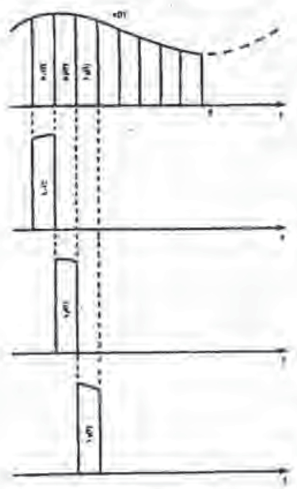
To view convolution in action, we can take a series of snapshots of the terms present at five consecutive sample times:

| t=0T | t=1T | t=2T | t=3T | t=4T |
|---|---|---|---|---|
| $x_0h_0$ | $x_0h_1$ | $x_0h_2$ | $x_0h_3$ | $x_0h_4$ |
|  | $x_1h_0$ | $x_1h_1$ | $x_1h_2$ | $x_1h_3$ |
|  |  | $x_2h_0$ | $x_2h_1$ | $x_2h_2$ |
|  |  |  | $x_3h_0$ | $x_3h_1$ |
|  |  |  |  | $x_4h_0$ |

The response is the sum of the terms in each column:

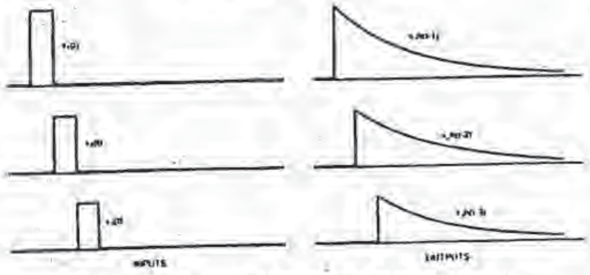
$$y_0 = x_0h_0$$
$$y_1 = x_0h_1 + x_1h_0$$

Fig. 13-6.
A graphical representation of convolution (from Blesser)



(A) Convolution may be viewed as an averaging or smoothing operation in which a waveform is considered as discrete pieces.

(B) Each piece produces an output response.



(C) The overall response is the summation of the individual responses.

$$y_2 = x_0h_2 + x_1h_1 + x_2h_0$$
$$y_3 = x_0h_3 + x_1h_2 + x_2h_1 + x_3h_0$$
$$y_4 = x_0h_4 + x_1h_3 + x_2h_2 + x_3h_1 + x_4h_0$$

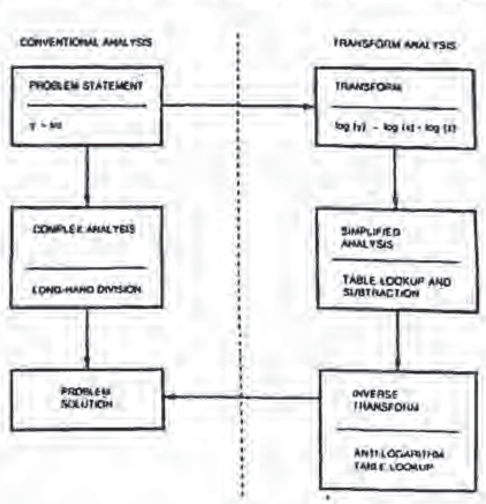
To find the convolved response we would reverse the impulse response, and align $h_0$ with the current $x$ sample to generate the ordered weighted product. The rest of the sequence would be obtained by moving the reversed impulse response until it has passed through the duration of the samples of interest, be it finite or infinite in length.

More generally, when two waveforms are multiplied together, their spectra are convolved, and if two spectra are multiplied, their determining waveforms are multiplied. The response to any input waveform can be determined from the impulse response of the network, and its response to any part of the input waveform. As noted, the convolution of two signals in the time domain corresponds to multiplication of their Fourier transforms in the frequency domain (as well as the dual correspondence). The trick is that any signal can be considered to be a sum of impulses.

## The Mathematical Transform

A transform is a mathematical tool used to simplify a solution to a problem. Transforms are used extensively, for example, in digital filter design. One example of a transform is a logarithm; figure 13-8 shows the relationship between a conventional and transform analysis. The problem is to determine

Fig. 13-8.
An example of the use of a transform, comparing long division with the use of logarithms



the quotient $y = x/z$. If high accuracy is required, a conventional (manual) solution dictates long-hand division, a time consuming process. Using a transform, we choose logarithms to substitute subtraction for division. The solution only requires us to look up the log (x) and log (y) in a table of logarithms, subtract, and then use an inverse transform, the antilog of log (y) to complete the solution.

In general, transforms result in a simplified problem solving analysis. Often, a digital implementation can be derived from an existing analog representation. For example, a stable analog filter may be described by the system function H(s). Its frequency response is found by evaluating H(s) at points on the imaginary axis of the s-plane. In the function H(s), s can be replaced by a rational function of z, which maps the imaginary axis of the s-plane onto the unit circle of the z-plane. The resulting system function H(z) is evaluated along the unit circle and takes on the same values of H(s) evaluated along its imaginary axis.

These mapping functions between the two planes are performed in various ways. Two important transforms used in DSP are the z-transform and the Fourier transform for discrete signals (FTD). The FTD generates a continuous spectrum but is difficult to compute. Thus, a sampled spectrum for discrete time signals of finite duration is implemented as the discrete Fourier transform (DFT). These approaches can be applied to all continuous and discrete time functions, with applications ranging from population growth to economic analysis, from chemical reactions to digital audio.

Just as the Fourier transform can generate the spectrum of a continuous signal, the DFT can be used to generate the spectrum of a discrete signal, expressed as a set of harmonically related sinusoids with unique amplitude and phase. The DFT takes samples of a waveform and operates on them as if they were an infinitely long waveform comprised of sinusoids, harmonically related to a fundamental frequency corresponding to the original sample period. An inverse DFT can recover the original sampled signal. The DFT is often generated with the fast Fourier transform (FFT), a fast and efficient algorithm for spectral computation. The FFT is not another type of transformation, but rather a method of calculating the DFT with fewer operations. In general, a number of short length DFTs are calculated, then the results are combined. However, the FFT can be applied to various calculation methods and strategies, including analysis of signals and filter design.

The FFT will transform a time series, such as the impulse response of a network, into the real and imaginary parts of the impulse response in the frequency domain. In this way, the magnitude and phase of the network's transfer function may be obtained. An inverse FFT can produce a time domain signal. FFT filtering is accomplished through multiplication of spectra. The impulse response of the filter is transformed to the frequency domain. Real and imaginary arrays, obtained by FFT transformation of overlapping segments of the signal, are multiplied by filter arrays, and an inverse FFT produces a filtered signal. The FFT can be efficiently computed; thus, it may be used as an alternative to time domain convolution if the overall number of multiplications is fewer. An example of an FFT application is given in the section on restoration, later in this chapter.

The DFT is essentially a special case of the z-transform. Whereas with the z-transform we may operate with any complex value, with the Fourier transform we operate with a particular complex value, $e^{j\omega}$. When $z = e^{j\omega}$, the z-transform is identical to the Fourier transform. The z-transform of a sequence x(n) is defined as:

$$X(z) = \sum_{n=-\infty}^{\infty} x(n) z^{-n}$$

where,

z is a complex variable.
$z^{-1}$ represents a time delay.

The z-transform functions for discrete signals in the same way that the Laplace transform functions for continuous signals. As opposed to the DFT used for literal operations, the z-transform is primarily a mathematical tool used in digital signal processing theory. For example, we could take the z-transform of the convolution equation, such that the z-transform of an input times the z-transform of a filter's impulse response is equal to the z-transform of the filter's output. In other words, the ratio of the filter output transform to the filter input transform is the transform of the impulse response. Furthermore, this ratio H(z) is a fixed function determined by the filter. The Fourier transform of a discrete signal corresponds to the z-transform on the unit circle in the z-plane.

## Poles and Zeros

Mathematically, we can state that the transfer function H(z) of a linear, time-invariant discrete-time filter is defined to be the z-transform of the impulse response h(n). The spectrum of a function is equal to the z-transform evaluated on the unit circle. The transfer function of a digital filter can be written in terms of its z-transform; this permits analysis in terms of the filter's poles and zeros. The roots of the numerator's polynomial of the transfer function are the zeros of the filter, and the denominator's roots are its poles. Mathematically, zeros make $H(z) = 0$, and poles make H(z) nonanalytic. When the magnitude of H(z) is plotted as a function of z, poles appear at a distance above the z-plane and zeros touch the z-plane. One might imagine the flat z-plane and above it a contour, the magnitude transfer function, passing through the poles and zeros—with peaks on top of poles, and valleys centered on zeros. Tracing the rising and falling of the contour around the unit circle yields the frequency response. For example, the gain of a filter at any frequency can be measured by the magnitude of the contour.

If we plot $|z| = 1$ on the complex plane we obtain the unit circle; $|z| > 1$ specifies all points on the complex plane that lie outside the unit circle, and $|z| < 1$ specifies all points inside it. The z-transform of a sequence can be represented by plotting the locations of the poles and zeros on the complex plane. The z-transform has an inverse transform, often obtained through partial fraction expansion.

## This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

## BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

☐ BLACK BORDERS

☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES

☐ FADED TEXT OR DRAWING

☑ BLURRED OR ILLEGIBLE TEXT OR DRAWING

☐ SKEWED/SLANTED IMAGES

☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS

☐ GRAY SCALE DOCUMENTS

☑ LINES OR MARKS ON ORIGINAL DOCUMENT

☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY

☐ OTHER: ______________________

**IMAGES ARE BEST AVAILABLE COPY.**
**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**

*Proceedings of the*

# International Conference on Multimedia Computing and Systems

**June 17 – 23, 1996** **Hiroshima, Japan**

*Sponsored by*
The IEEE Computer Society Technical Committee on Multimedia Computing

IEEE Computer Society Press
Los Alamitos, California

Washington • Brussels • Tokyo

# Digital Watermarks for Audio Signals

*Laurence Boney*[2] *Ahmed H. Tewfik*[1] *Khaled N. Hamdy*[1]

[1] Department of Electrical Engineering, University of Minnesota, Minneapolis, MN 55455
[2] Departmente Signale, ENST, Paris, France 75634
Email: boney@email.enst.fr, tewfik@ee.umn.edu, khamdy@ee.umn.edu

## ABSTRACT

*In this paper, we present a novel technique for embedding digital "watermarks" into digital audio signals. Watermarking is a technique used to label digital media by hiding copyright or other information into the underlying data. The watermark must be imperceptible or undetectable by the user and should be robust to attacks and other types of distortion. In our method, the watermark is generated by filtering a PN-sequence with a filter that approximates the frequency masking characteristics of the human auditory system. It is then weighted in the time domain to account for temporal masking. We discuss the detection of the watermark and assess the robustness of our watermarking approach to attacks and various signal manipulations.*

## 1. INTRODUCTION

In today's digital world, there is a great wealth of information, which can be accessed in various forms: text, images, audio, and video. It is easy to ensure the security of "analog documents" and protect the author (author will be used to also denote composer, artist, designer, etc.) from having his work stolen or copied. For example, a painting is signed by the artist, books and albums have copyright labels imprinted inside the cover. The question is *how do you copyright or label digital information and preserve its security without destroying or modifying the content of the information.*

One approach to data security is to use cryptographic techniques. In cryptology, the information is scrambled using an encryption transformation before it is sent and the information can be viewed after de-scrambling with the inverse transformation. A public-key cryptosystem can be used to implement an electronic mail system in which messages are kept private and can be signed [1]. The security of the encryption algorithm is based on the fact that no one has discovered an algorithm which can factor composite numbers with two very large prime factors (on the order of 200 digits) in a *reasonable amount of time*. Note that cryptosystems restrict access to the document and do not label or stamp them. Once the documents are decrypted, the "signature" is removed and there is *no proof of ownership* such as a label, stamp, or watermark [2]. Cryptology, as discussed in [3], may be used for digital TV broadcasting to provide conditional access for pay TV, watermarking of images for copyright protection, and image signature for authentication. Note, that it is useful to consider the binary representation of these large numbers and their prime factors as codewords for the signatures.

Data hiding, or steganography, refers to techniques for embedding watermarks, signatures, tamper protection, and captions in digital data. Captioning is an application which requires a large amount of data. However, it need not be invariant to removal because it contains extra non-critical information which may be of benefit to the author and the user. On the other hand, watermarking is an application which embeds the least amount of data, but requires the greatest robustness because the watermark is required for copyright protection [4]. Note that data hiding does not restrict access to the original information as does encryption.

A watermark, or an invisible stamp, could be used to provide proof of "authorship" of a signal. Similarly, a signature is used to provide proof of ownership and track illegal copies of the signal. The watermark must be embedded in the data such that it is *imperceptible* by the user [3, 4, 5]. Moreover, the watermark should:

- **be inaudible**: the watermark should not affect the audio quality of the original signal;
- **be statistically invisible**: a "pirate" should not be able to detect the watermark by comparing several signals belonging to the same author to prevent unauthorized detection and/or removal;
- **have similar compression characteristics as the original signal** to survive compression/decompression operations ;
- **must be robust to deliberate attacks by "pirates"**;
- **must be robust to standard signal manipulation and processing operations** on the host data, e.g., filtering, resampling, compression, noise, cropping, A/D-D/A conversions, etc;
- **should be embedded directly in the data, not in a header** to prevent removal;
- **must support multiple watermarkings** for wider applicability;
- **should be self-clocking** for ease of detection in the presence of cropping and time-scale change operations.

---

*THIS WORK WAS PARTIALLY SUPPORTED BY AFOSR UNDER GRANT AF/F49620-94-1-0461 AND BY NSF UNDER GRANT NSF UNDER GRANT NSF/INT-9406954.

0-8186-7436-9/96 $5.00 © 1996 IEEE
*Proceedings of MULTIMEDIA '96*

Observe that a "pirate" can defeat a watermarking scheme in two ways. He may manipulate the audio signal to make the watermark undetectable. Alternatively, he may establish that the watermarking scheme is unreliable, e.g., that it produces too many false alarms by detecting a watermarks where none is present. Both goals can be achieved by adding inaudible jamming signals to the audio piece. Therefore, the effectiveness of a watermarking scheme must be measured by its ability to detect a watermark when one is present (probability of detection) *and* the probability that it detects a watermark when none is present (probability of a false alarm) in the presence of jamming signals and signal manipulations.

Several techniques for data hiding in images have been developed. Two methods for watermarking images are proposed in [6]. The first approach embeds a PN-sequence on the least significant bit (LSB) of the data. This provides easy and rapid decoding of the watermark or signature. In the second approach, a PN-sequence (watermark) is added to the LSB of the data. This is more difficult to decode, providing more security. As with any approach which modifies the LSB of the data, however, these watermarks are highly sensitive to noise and are easily corrupted.

In other coding schemes, the watermarks are made to appear as quantization noise as they are embedded into the images [7, 8]. The first method uses a predictive coding scheme to embed the watermark into the image. In the second method, the watermark is embedded into the image by dithering the image based on the statistical properties of the image. This scheme is not robust to attacks such as requantization and cropping.

In [9], a watermark for an image is generated by modifying the luminance values inside 8x8 blocks of pixels, adding one extra bit of information to each block. The choice of the modified block is secretly made by the encoder. In [10], a 2-D signature is generated and is embedded into the image by modifying the intensity levels of the image, whose corresponding signature pixels is one. A method using a JPEG model based, frequency hopped, randomly sequenced pulse position modulated code in [11] is robust to operations such as lossy data compression, lowpass filtering, and color space conversion. The watermarking problem is viewed as a problem in digital communications in [12]: a codeword is generated and used to modulate selected coefficients of the DCT or wavelet transform of a block in an image.

Data may be hidden in images by exploiting the properties of the human visual system (HVS), such as sensitivity to contrast as a function of spatial frequency, the masking effect of edges, and sensitivity to changes in gray-scale [4]. In [4, 13], techniques for data hiding in images are discussed. The first, an LSB method called "Patchwork," is a statistical technique which randomly chooses $n$ pairs $(a_i, b_i)$ of points in an image and increases the brightness of $a_i$ by one unit while simultaneously decreasing the brightness of $b_i$. The second, texture block coding, hides data by mapping a random texture pattern in an image to another region in the image with a similar texture pattern. This method is limited to images that possess large areas of random texture. In [13], an encoding scheme is made resistant to affine transformations (scaling, translations, rotations) by embedding crosses in an image. Xerox DataGlyph technology [4, 14] adds a barcode to its images according to a predetermined set of geometric modifications. In [15] data is hidden in the chrominance signal of NTSC by exploiting the HVS temporal over-sampling of color. Adelson [16] proposes a scheme that embeds digital data into analog TV signals. The method substitutes high-spatial frequency image data for "hidden" data in a pyramid-encoded image. However, the scheme is particularly susceptible to filtering and rescaling.

A method similar to ours is proposed in [5], where the $N$ largest frequency components of an image are modified by Gaussian noise. However, the scheme only modifies a subset of the frequency components and does not take into account the HVS. The audio watermark we propose here embeds the maximum amount of information throughout the spectrum while still remaining perceptually inaudible. It is well-known that detection performance (i.e., the probability of detection and the probability of false alarm) improves with the energy of the signal to be detected. Therefore, we effectively improve the performance of the watermarking scheme by increasing the energy of the watermarked signal while keeping it inaudible.

Data hiding techniques have also been applied to audio signals [4, 13]. In Direct Sequence Spread Spectrum Coding (DSSS), the signature, a binary codeword, is modulated by both a PN-sequence and the audio signal using bi-phase shift keying. It is then added to the original signal as additive random noise. The perceivable noise added to the signal can be reduced by adaptive coding and redundant coding. In Phase Coding, binary information is embedded in the audio signal by modifying the phases of each frequency component of the Discrete Short Time Fourier Transform of the signal. Because the human auditory system (HAS) is not highly sensitive to phase distortion, the data produce no audible distortion.

In this paper, we present a novel technique for embedding digital watermarks into audio signals. The watermark is generated by filtering a PN-sequence with a filter that approximates the frequency masking characteristics of the HAS. It is then weighted in the time domain to account for temporal masking. Note that our approach is similar to that of [4, 13] in that we shape the frequency characteristics of a PN-sequence. However, unlike [4, 13] we use perceptual masking models of the HAS to generate the watermark. In particular, our scheme for audio is the only one that uses the frequency masking models of the HAS along with the temporal masking models to hide the copyright information in the signal.

We also provide a study of the detection performance of our watermarking scheme. Our results indicate that our scheme is robust to lossy coding/decoding, D/A - A/D conversion, signal resampling, and filtering. We are currently studying its robustness to time-frequency changes.

Finally, observe that the approach described here for watermarking audio signals can also be used to watermark image and video data with appropriate modifications and extensions (c.f. [17], [18]).

## 2. BACKGROUND

### 2.1. Masking

Masking is the effect by which a faint but audible sound becomes inaudible in the presence of another louder audible sound, masker [19]. The masking effect depends on the both spectral and temporal characteristics of both the masked signal and the masker [19]. Frequency masking refers to masking which occurs in the frequency domain. If two signals which occur simultaneously are close together in frequency, the stronger masking signal will make the weaker masked signal inaudible. The masking threshold of a masker depends on the frequency, sound pressure level (SPL), and tone-like or noise-like characteristics of both the masker and the masked signal [20]. It is easier for a broadband noise to mask a tonal, than for a tonal signal to mask out a broadband noise. Moreover, higher frequency signals are more easily masked. Temporal masking refers to both pre- and post-masking. Pre-masking effects render weaker signals inaudible before the stronger masker is turned on, and post-masking effects render weaker signals inaudible after the stronger masker is turned off. Pre-masking occurs from 5-20 msec. before the masker is turned on while post-masking occurs from 50-200 msec. after the masker is turned off [20].

Using the frequency masking information of the HAS, we can shape the spectral characteristics of the watermark. Processing of impulsive signals such as castanets can cause audible pre-echoes. Similarly, we can use temporal masking information to eliminate these effects.

### 2.2. Frequency Masking: MPEG-1 Psychoacoustic Model

Audio signals consist of telephone quality speech, wideband speech, and wideband audio. The frequency ranges for these types of audio signals are 300-3400 Hz for telephone speech signals, 50-7000 Hz for wideband speech, and 20-20000 Hz for high quality wideband audio. The human ear acts as a frequency analyzer and can detect sounds with frequencies which vary from 10 Hz to 20000 Hz. The HAS can be modeled by a set of 26 bandpass filters with bandwidths that increase with increasing frequency. The 26 bands are known as the critical bands. The critical bands are defined around a center frequency in which the noise bandwidth is increased until there is just noticeable difference in the tone at the center frequency. Thus if a faint tone lies in the critical band of a louder tone, the faint tone will not be perceptible.

Frequency masking models have already been defined for the perceptual coding of audio signals because it is not necessary to code perceptually irrelevant information. In this work, we use the masking model defined in MPEG Audio Psychoacoustic Model 1, for layer I [21]. The masking method is summarized as follows for a 32 kHz sampling rate [21, 22]. The MPEG model also supports sampling rates of 44.1 kHz and 48 kHz.

- First Step: *Calculate the Spectrum*
  Each 16 ms segment of the signal $s(n)$, N=512 samples, is weighted with a Hann window, $h(n)$:

$$h(n) = \frac{\sqrt{8/3}}{2}[1 - \cos(2\pi \frac{n}{N})] \qquad (1)$$

The power spectrum of the signal $s(n)$ is calculated as follows:

$$S(k) = 10 * \log_{10}[\frac{1}{N}\|\sum_{n=0}^{N-1} s(n)h(n)\exp(-j2\pi\frac{nk}{N})\|^2] \qquad (2)$$

The maximum is normalized to a reference sound pressure level of $96dB$.

- Second Step: *Identify Tonal Components*
  Tonal (sinusoidal) and non tonal (noisy) components are identified because their masking models are different.
  A tonal component is a local maximum of the spectrum ( $S(k) > S(k+1)$ et $S(k) \geq S(k-1)$ ) satisfying:

$$S(k) - S(k+j) \geq 7dB$$
$$j \in [-2, +2] \text{ if } 2 < k < 63$$
$$j \in [-3, -2, +2, +3] \text{ if } 63 \leq k < 127$$
$$j \in [-6, ..., -2, +2, ..., +6] \text{ if } 127 \leq k \leq 250$$

We add to its intensity those of the previous and following component. Other tonal components in the same frequency band are no longer considered.
Non-tonal components are made of the sum of the intensities of the signal components remaining in each of the 24 critical bands between 0 and 15500 Hz. (The auditory system behaves as a bank of bandpass filters, with continuously overlapping center frequencies. These "auditory filters" can be approximated by rectangular filters with critical bandwidth increasing with frequency. In this model, the audible band is therefore divided into 24 non-regular critical bands.)

- Third Step: *Remove Masked Components*
  Those components below the absolute hearing threshold and tonal components separated by less than 0.5 Barks.

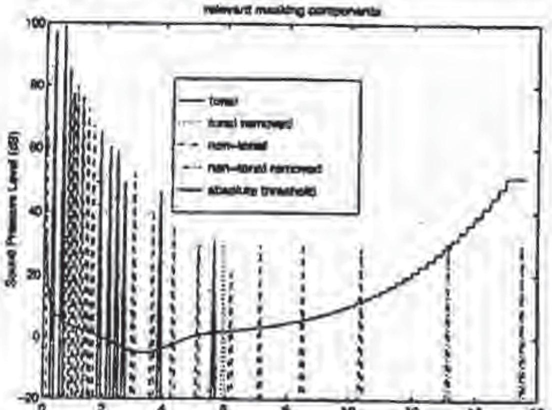


**Figure 1. Second and Third step in generation of masking threshold**

- Fourth Step: *Individual and Global Masking Thresholds*

In this step, we account for the frequency masking effects of the HAS. We need to discretize the frequency axis according to hearing sensitivity and express frequencies in Barks. Note that hearing sensitivity is higher at low frequencies. The resulting masking curves are almost linear and depend on a masking index different for tonal and non-tonal components. They are characterized by different lower and upper slopes depending on the distance between the masked and the masking component. We use $f_1$ to denote the set of frequencies present in the test signal. The global masking threshold for each frequency $f_2$ takes into account the absolute hearing threshold $S_a$ and the masking curves $P_2$ of the $N_t$ tonal components and $N_n$ non-tonal components:

$$S_m(f_2) = 10 * \log_{10}[10^{S_a(f_2)/10} + \sum_{j=1}^{N_t} 10^{P_2(f_2,f_1,P_1)/10} + \sum_{j=1}^{N_n} 10^{P_2(f_2,f_1,P_1)/10}] \quad (3)$$

The masking threshold is then the minimum of the local masking threshold and the absolute hearing threshold in each of the 32 equal width subbands of the spectrum. Any signal which falls below the masking threshold is inaudible.

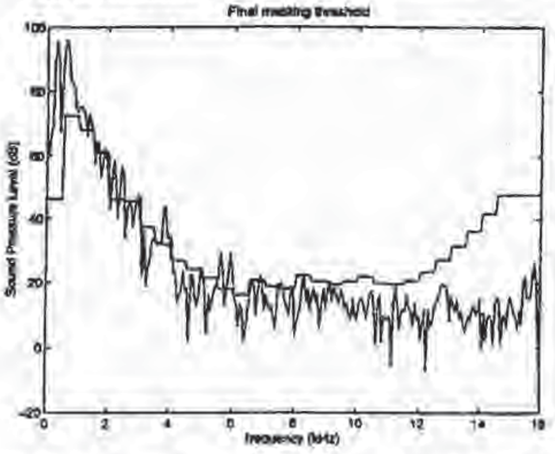


**Figure 2. Fourth step in generation of masking threshold**

### 2.3. PN-sequences

PN-sequences form the basis of our watermarking scheme because of their noise-like characteristics, resistance to interference, and good auto-correlation properties. Spread spectrum communication systems use pseudo-noise (PN) sequences to modulate transmitted data into noise-like wideband signals so they blend into the background [23]. Spread spectrum signals are resistant to interference such as unintentional interference, channel noise, multiple users, multipath interference, or intentional jammers [23].

PN-sequences are periodic noise-like binary sequences generated by feedback shift register of fixed length $m$ [23]. The feedback is linear, that is, it consists of only modulo-2 adders. This prevents the zero state from occurring, which provides an output of only zeros. The maximum period of a PN-sequence is $N = 2^m - 1$ [23]. The feedback connections for maximal length PN-sequences with m varying from 1 to 89 are provided in [24].

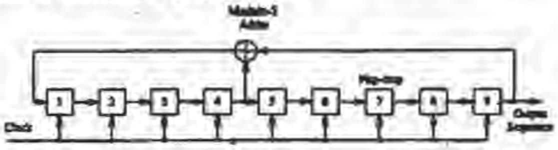


**Figure 3. Shift register with m=9, N=511**

Maximum length PN-sequences, also called m-sequences, are used in our watermarking scheme because they provide an easy way to generate a unique code for an author's identification. Moreover, like random binary sequences, m-sequences have 0's and 1's occur with equal probabilities. Also, the number of 1's is always one greater than the number of 0's. M-sequences also have good autocorrelation properties [23]: the autocorrelation function (ACF) has period N and it is binary valued. The ACF has peaks equal to 1 at 0, N, 2N, etc. and is approximately $1/N$ elsewhere. Because of these periodic peaks, the m-sequence is self-clocking. This allows the author to synchronize with the embedded watermark during the detection process. This is important if the signal is cropped and resampled.

## 3. WATERMARK DESIGN

Each audio signal is watermarked with a unique codeword. Our watermarking scheme is based on a repeated application of a basic watermarking operations on processed versions of the audio signal. The basic method uses three steps to watermark an audio segment as shown in 4. The complete watermarking scheme is shown in 5. Below we provide a detailed explanation of the basic watermarking step and the complete watermarking technique.

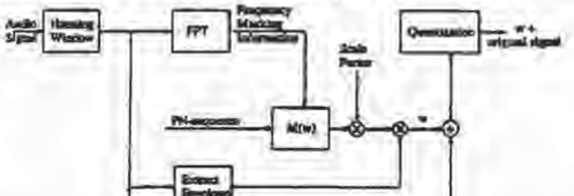


**Figure 4. Watermark Generator: First stage for audio**

### 3.1. The basic watermarking step

The basic watermarking step starts with a PN-sequence. To generate the watermark, we first calculate the masking threshold of the signal using the MPEG Audio Psychoacoustic Model 1, as described above. The masking threshold

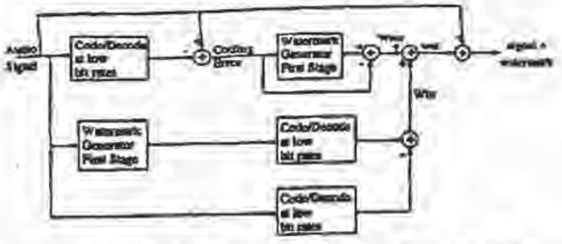


Figure 5. Full Watermark Generator for audio

is determined on consecutive audio segments of 512 samples. Each segment is weighted with a Hanning window. Consecutive blocks overlap by 25%.

The masking threshold is then approximated with a $10^{th}$ order all-pole filter, $M(\omega)$, using a least squares criterion. The PN-sequence $seq(\omega)$, is filtered with the approximate masking filter, $M(\omega)$, in order to ensure that the spectrum of the watermark is below the masking threshold, as shown in Fig. 6. In this example, we used an m-sequence with $m = 9$.

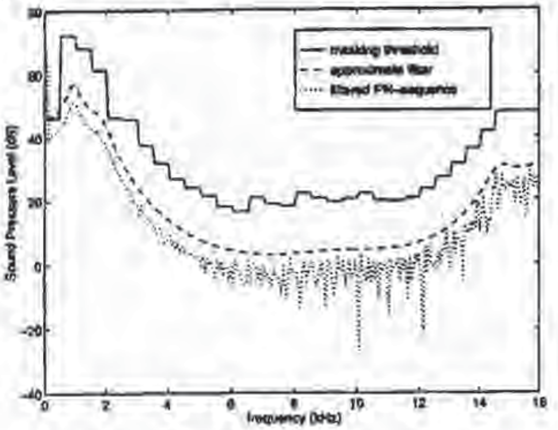


Figure 6. Filtered PN-sequence

Since the spectral content of the audio signal changes with time, watermarks added to different blocks will be in general different even if they are generated from the same starting PN-sequence. However, we still use different PN-sequences for different blocks to make the statistical detection by an unauthorized user of the watermark more difficult. Note also that using long PN-sequences or embedding long cryptographic digital signatures also helps in that respect.

Frequency domain shaping is not enough to guarantee that the watermark will be inaudible. Frequency domain masking computations are based on a Fourier transform analysis. A fixed length Fourier transform does not provide good time localization for our application. In particular, a watermark computed using frequency domain masking will spread in time over the entire analysis block. If the signal energy is concentrated in a time interval that is shorter than the analysis block length, the watermark is not masked outside of that subinterval. This then leads to audible distortion, e.g., pre-echoes. To address this problem, we weight the watermark in the time domain with the relative energy of the signal. Specifically, denote by $s(n)$ the $n$th sample of the audio signal. We modify the watermark, $w(n)$ as follows

$$w(n) = w(n) * \frac{s(n)^2}{\sum_{k=1}^{N} s(k)^2}. \tag{4}$$

The time domain weighting operation described above attenuates the energy of the computed watermark. In particular, watermarks obtained as above have amplitudes that are typically smaller than the quantization step size. Therefore, the watermark would be lost during the quantization process. Note also that, as observed earlier, detection performance is directly proportional to the energy of the watermark. We have found that it is possible to prevent watermark loss during quantization and improve detection performance by amplifying the watermark by 40 dB before weighting it in the time domain with the relative energy of the signal. In most cases, this amplification does not affect the audibility of the watermark because of the attenuation effect of the time domain weighting operation. However, to guarantee inaudibility, we re-check that the final watermark falls below the masking threshold in the frequency domain. If the amplitude of the watermark at a given frequency exceeds the masking threshold at that frequency we simply reduce it to the maximum allowable level.

### 3.2. The full watermarking scheme

As mentioned above, the watermarking scheme must be robust to coding operations. Low bit rate audio coding algorithms tend retain only the low frequency information in the signal. We therefore need to guarantee that most of the energy of the watermark lies in low frequencies. After experimenting with many schemes, we have found that the best way to detect the low frequency watermarking information is to generate a low-frequency watermark as the difference between a low bit rate coded/decoded watermarked signal and the coded/decoded original signal at the same bit rate. Watermarking is done using the basic watermarking step described above. The low bit rate chosen to implement this operation is the minimal bit rate for which near-transparent audio coding is known to be possible for signals sampled at the rate of the original signal. For signals sampled at 44.1 kHz, the watermark is generated using a bit rate of 64 kbits/sec. For signals sampled at 32 kHz, the watermark is generated using a bit rate of 48 kbits/sec. This scheme is more effective than other schemes that attempt to add the watermark on a lowpass filtered version of the signal because the coding/decoding operation is not a linear and does not permute with the watermarking operation.

Fig. 5 illustrates the above procedure for signals sampled at an arbitrary sampling rate. The low-frequency watermarking signals is shown as $w_{br}$ in Fig. 5. It is given by

$$w_{br} = (watermark_{firststage})_{br} - (originalsignal)_{br} \tag{5}$$

Here, the subscript $br$ refers to the bit rate of the coder/decoder.

For best watermark detection performance at higher bit rates, we need to add watermarking information in the higher frequency bands. We do so by producing a watermark $w_{err}$ for the coding error signal that is the difference between the original audio signal and its low bit rate coded version:

$$codingerror = (originalsignal) - (originalsignal)_{br}. \quad (6)$$

The watermark $w_{err}$ is computed using the basic watermarking step described at the beginning of this section.

The final watermark is the sum of the low-frequency watermark and the coding error watermark:

$$wat = w_{br} + w_{err}. \quad (7)$$

### 3.3. Experimental testing of the audibility of the watermarks

We used segments of four different musical pieces as test signals throughout the experiment: the beginning of the third movement of the sonata in B flat major D 960 of Schubert, interpreted by Vladimir Ashkenazy, a castanet piece, a clarinet piece, and a segment of "Tom's Diner" an a *capella* song by Suzanne Vega (svega). The Schubert signal is sampled at 32 kHz. All other signals are sampled at 44.1 kHz. Note that the castanets signal is one of the signals prone to pre-echoes. The signal svega is significant because it contains noticeable periods of silence. The watermark should not be audible during these silent periods.

The quality of the watermarked signals was evaluated through informal listening tests. In the test, the listener was presented with the original signal and the watermarked signal and reported as to whether any differences could be detected between the two signal. Eight people of varying backgrounds, including the authors, were involved in the listening tests. One of the listeners has the ability to perceive absolute pitch and two of the listeners have some background in music.

In all four test signals, the watermark introduced no audible distortion. No pre-echoes were detected in the watermarked castanet signal. The quiet portions of svega were similarly unaffected.

## 4. DETECTION OF THE WATERMARK

Let us now describe the watermark detection scheme and the detection results that we have obtained. In the experimental work described below, we used shaped inaudible noise to simulate attacks by pirates and distortions due to coding. We also tested the effects of filtering, coding, D/A - A/D converting and re-sampling on the detection performance of the proposed scheme. The detection results that we report below are based on processing 100 blocks of the observed signal of 512 samples. Note that this corresponds to 1.6 sec at the 32 kHz sampling rate and 1.16 sec at the 44.1 kHz sampling rate.

Our detection scheme assumes that the author has access to the original signal and the PN-sequence that he used to watermark the signal. It also assumes that the author has computed the approximate bit rate of the observed audio sequence $r(k)$. To decide whether the given signal $r(k)$ has been watermarked or not, the author subtracts from $r(k)$ a coded version $s_{br}$ of the original audio signal $s(k)$. The signal $s_{br}$ is produced by coding $s(k)$ at the estimated bit rate of $r(k)$ using the MPEG coding procedure. Note that $r(k)$ itself may have been coded using a different coding algorithm. The difference between the output of the MPEG coding algorithm operating on the original signal at the estimated bit rate and that of the actual coding algorithm at the true bit rate will appear as an additive noise signal.

Next, the author needs to solve the following hypothesis testing problem:

- $H_0 : x(k) = r(k) - s_{br}(k) = n(k)$
- $H_1 : x(k) = r(k) - s_{br}(k) = w'(k) + n(k)$.

Here, $n(k)$ denotes an additive noise process that includes errors due to different coding algorithms and signal manipulations, intentional jamming signals and transmission noise. The signal, $w'(k)$, is the modified watermark. Since the precise nature of $n(k)$ is unknown, we solve the above hypothesis testing problem by correlating $x(k)$ with $w'(k)$ and comparing the result with a threshold. Note that one needs to estimate time-scale modifications prior to correlations if such modifications have been performed on the signal. Fig. 7 shows the result of correlating a watermark corresponding to a segment of the Schubert audio piece with itself, the jammed watermark corrupted by frequency shaped noise of maximum masked intensity and shaped noise of maximum masked intensity alone. In all cases, the signal was not coded. The figure clearly indicates that reliable detection is feasible.

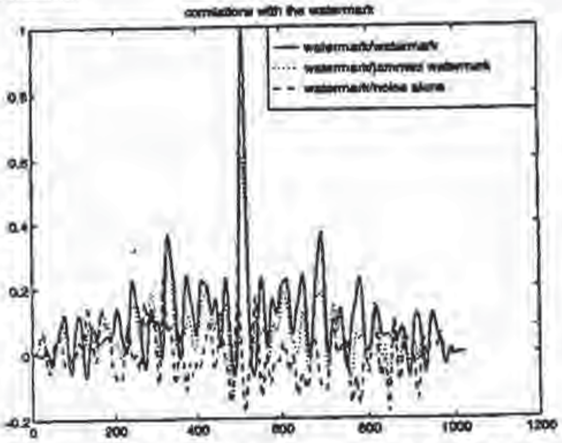


Figure 7. Detection of the watermark in Schubert with additive noise

### 4.1. Generation of the Additive Noise

Noise which has the same spectral characteristics as the masking threshold provides an approximation of the worst possible additive distortion to the watermark. This type of distortion is a good worst case model for distortions due to intentional jamming with inaudible signals and mismatches between the actual and assumed coding algorithms.

The noise that we have used in our experiments was generated in the same way as the watermark. Specifically, the masking threshold is first shifted $+40dB$ and multiplied by the discrete Fourier transform of a Gaussian white noise process. The resulting noise is then weighted in time by the relative energy of the signal. After quantization, we filter this shaped noise by the masking threshold and requantize it. The resulting noise is almost completely inaudible and is a good approximation of the maximum noise that we can add below the masking threshold, as shown in Fig. 8.

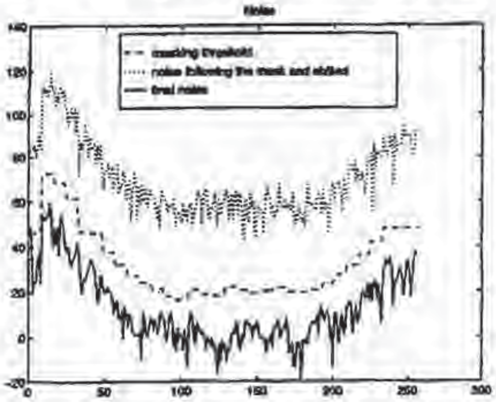


**Figure 8.** final noise after weighting in time, requantization, filtering by the mask and last requantization

### 4.2. Summary of Detection Results

Let us now summarize the detection results that we have obtained. Each group of results is meant to illustrate the robustness of our approach to a specific type of signal manipulation.

*Robustness to coding*
To test the robustness of our watermarking approach to coding, we added noise to several watermarked and non-watermarked audio pieces and coded the result. The noise was almost inaudible and was generated using the technique described above. We then attempted to detect the presence of the watermark in the decoded signals.

The coding/decoding was performed using a software implementation of the ISO/MPEG-1 Audio Layer III coder [25] with several different bit rates (64 kbits/s, 128 kbit/s, 160 kbit/s, 224 kbit/s and 320 kbit/s).

Table 1 below gives the probabilities of detection and false alarm for the final watermark in the following signals: the Schubert signal, a castanet signal, and a clarinet signal. Note that the probability of detection of the watermark, $P_{detect}$, is 1 or nearly 1 in all cases. Equally important, the probability of false alarm, $P_{falsealarm}$ is nearly 0 in all cases.

*Robustness to multiple watermarking*
There are many instances where it is useful to add multiple watermarks to a signal. For example, there may be multiple authors/composers for a piece of music, each with his/her own unique Id. When detecting a specific watermark, the other watermarks are considered to be noise.

Tables 2, 3, 4, and 5 summarize watermark detection results in the presence of other watermarks. Several signals containing three watermarks were corrupted by the coding/decoding operation. The detection was performed using each of the three watermarks. Again, note that $P_{detect}$ is 1 or nearly 1 and that $P_{falsealarm}$ is 0 or nearly 0 in all cases.

*Robustness to resampling*
Our experiments also indicate that the proposed watermarking scheme is robust to signal resampling. Specifically, the watermarked signal is resampled and then corrupted by the coding/decoding operation. For a threshold of 0.68, $P_{detect}$ is 1 and $P_{falsealarm}$ is 0 at all 5 coder/decoder bit rates.

We are currently assessing the robustness of our scheme to time-scale modifications of the signal.

## 5. CONCLUSIONS

Our method for the digital watermarking of audio signals extends the previous work on images. Our watermarking scheme consists of a maximal length PN-sequence filtered by the approximate masking characteristics of the HAS and weighted in time, our watermark is imperceptibly embedded into the audio signal and easy to detect by the author thanks to the correlation properties of PN-sequences. Our results show that our watermarking scheme is robust in the presence of additive noise, lossy coding/decoding, multiple watermarks, resampling, and time-scaling.

## REFERENCES

[1] R. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[2] E. Koch, J. Rindfrey, and J. Zhao, "Copyright protection for multimedia data," *Proc. of the Intl. Conf. on Digital Media and Electronic Publishing (6-8 Dec. 1994, Leeds, UK)*.

[3] B. Macq and J.-J. Quisquater, "Cryptology for digital tv broadcasting," *Proc. of IEEE*, pp. 944–957, June 1995.

[4] W. Bender, D. Gruhl, and N. Morimoto, "Techniques for data hiding," *Proc. of the SPIE*, 1995.

[5] I. Cox, J. Kilian, T. Leighton, and T. Shamoon, "Secure Spread Spectrum Watermarking for Multimedia." Tech. Rep. 95-10, NEC Research Institute, 1995.

[6] R. V. Schyndel, A. Z. Tirkel, and C. Osborne, "A digital watermark," in *Proc. of ICIP'94*, vol. II, pp. 86–90, November 1994.

[7] K. Matsui and K. Tanaka, "Video steganography: How to secretly embed a signature in a picture," in *IMA Intellectual Property Project Proc. 1:1*, Jan. 1994.

[8] K. Tanaka, Y. Nakamura, and K. Matsui, "Embedding secret information into a dithered multi-level image," in *MILCOM 90: A New Era. 1990 IEEE Military Communications Conference*.

[9] O. Bruyndonckx, J.-J. Quisquater, and B. Macq, "Spatial method for copyright labeling of digital images," in *Nonlinear Signal Processing Workshop, Thessaloniki, Greece*, pp. 456–459, 1995.

[10] I. Pitas and T. Kaskalis, "Applying signatures on digital images," in *Nonlinear Signal Processing Workshop, Thessaloniki, Greece*, pp. 460–463, 1995.

[11] E. Koch and J. Zhao, "Towards robust and hidden image copyright labeling," in *Nonlinear Signal Processing Workshop, Thessaloniki, Greece*, pp. 452–455, 1995.

[12] F. Boland, J. O. Ruanaidh, and C. Dautzenberg, "Watermarking digital images for copyright protection," *IEE Intl. Conf. on Image Proc. and Its Apps., Edinburgh, 1995.*

[13] D. Gruhl, N. Morimoto, and W. Bender, "The data hiding homepage," *http://nif.www.media.mit.edu/DataHiding/index.html*, 1995.

[14] J. Gruber, "Smart paper," *Wired*, vol. 2, Dec. 1994.

[15] A. Lippman, "Receiver-compatible enhanced definition television system." U. S. Patent 5,010,405, 1991.

[16] E. Adelson, "Digital Signal Encoding and Decoding Apparatus." U. S. Patent 4,939,515, 1990.

[17] M. Swanson, B. Zhu, and A. Tewfik, "Transparent robust image watermarking," in *Submitted to ICIP'96*, (Lausanne, Switzerland), Sept. 1996.

[18] L. Boney, A. Tewfik, K. Hamdy, and M. Swanson, "Digital watermarks for multimedia." Submitted to U.S. Patent Office, February 1996.

[19] J. Johnston and K. Brandenburg, "Wideband coding–perceptual considerations for speech and music," in *Advances in Speech Signal Processing* (S. Furui and M. M. Sondhi, eds.), New York: Dekker, 1992.

[20] P. Noll, "Wideband speech and audio coding," *IEEE Comm. Mag.*, pp. 34–44, Nov. 1993.

[21] "Codage de l'image animee et du son associe pour les supports de stockage numerique jusqu'a environ 1,5 mbit/s," tech. rep., ISO/CEI 11172, 1993.

[22] N Moreau, *Techniques de Compression des Signaux*. Masson, 1995.

[23] S. Haykin, *Communication Systems, 3rd Edition*. John Wiley and Sons, 1994.

[24] R. Dixon, *Spread Spectrum Systems*. John Wiley and Sons, 1976.

[25] "Mpeg audio-layer 3 encoder/decoder software," 1994-1995. http://www.iis.fhg-de.

**Table 1.** Detection of watermark

| Bit Rate | Audio Signal | *svega* | *castanets* | *clarinet* |
|---|---|---|---|---|
| kbits/sec | Threshold | 0.38 | 0.58 | 0.79 |
| 64 | $P_{detect}$ | 1 | 0.9996 | 1 |
| | $P_{falsealarm}$ | 0 | 0.0007 | 0.0006 |
| 128 | $P_{detect}$ | 1 | 0.9996 | 1 |
| | $P_{falsealarm}$ | 0 | 0.0007 | 0 |
| 160 | $P_{detect}$ | 1 | 0.9978 | 1 |
| | $P_{falsealarm}$ | 0 | 0.0004 | 0 |
| 224 | $P_{detect}$ | 1 | 0.9978 | 1 |
| | $P_{falsealarm}$ | 0 | 0.0004 | 0 |
| 320 | $P_{detect}$ | 1 | 0.9996 | 1 |
| | $P_{falsealarm}$ | 0 | 0.0004 | 0 |
| | # of trials | 3000 | 2728 | 1571 |

**Table 2.** Multiple watermark detection on Schubert

| Bit Rate | Watermark | *Wat a* | *Wat b* | *Wat c* |
|---|---|---|---|---|
| kbits/sec | Threshold | 0.64 | 0.60 | 0.56 |
| 64 | $P_{detect}$ | 1 | 1 | 1 |
| | $P_{falsealarm}$ | 0 | 0 | 0 |
| 128 | $P_{detect}$ | 1 | 1 | 0.9920 |
| | $P_{falsealarm}$ | 0 | 0 | 0 |
| 160 | $P_{detect}$ | 1 | 1 | 1 |
| | $P_{falsealarm}$ | 0 | 0 | 0.001 |
| 224 | $P_{detect}$ | 1 | 1 | 1 |
| | $P_{falsealarm}$ | 0 | 0 | 0 |
| | # of trials | 1000 | 1000 | 1000 |

**Table 3.** Multiple watermark detection on castanets

| Bit Rate | Watermark | *Wat a* | *Wat b* | *Wat c* |
|---|---|---|---|---|
| kbits/sec | Threshold | 0.525 | 0.34 | 0.33 |
| 64 | $P_{detect}$ | 0.9970 | 1 | 1 |
| | $P_{falsealarm}$ | 0.1130 | 0 | 0 |
| 128 | $P_{detect}$ | 1 | 1 | 1 |
| | $P_{falsealarm}$ | 0 | 0 | 0 |
| 160 | $P_{detect}$ | 1 | 1 | 1 |
| | $P_{falsealarm}$ | 0 | 0 | 0 |
| 224 | $P_{detect}$ | 1 | 1 | 1 |
| | $P_{falsealarm}$ | 0 | 0 | 0 |
| | # of trials | 1000 | 1000 | 1000 |

**Table 4.** Multiple watermark detection on clarinet

| Bit Rate | Watermark | *Wat a* | *Wat b* | *Wat c* |
|---|---|---|---|---|
| kbits/sec | Threshold | 0.71 | 0.33 | 0.55 |
| 64 | $P_{detect}$ | 1 | 0.9920 | 1 |
| | $P_{falsealarm}$ | 0 | 0.006 | 0.002 |
| 128 | $P_{detect}$ | 1 | 1 | 1 |
| | $P_{falsealarm}$ | 0 | 0.002 | 0.015 |
| 160 | $P_{detect}$ | 1 | 1 | 1 |
| | $P_{falsealarm}$ | 0 | 0 | 0 |
| 224 | $P_{detect}$ | 1 | 1 | 1 |
| | $P_{falsealarm}$ | 0 | 0 | 0 |
| | # of trials | 1000 | 1000 | 1000 |

**Table 5.** Multiple watermark detection on svega

| Bit Rate | Watermark | *Wat a* | *Wat b* | *Wat c* |
|---|---|---|---|---|
| kbits/sec | Threshold | 0.61 | 0.57 | 0.60 |
| 64 | $P_{detect}$ | 1 | 1 | 1 |
| | $P_{falsealarm}$ | 0 | 0 | 0 |
| 128 | $P_{detect}$ | 1 | 1 | 1 |
| | $P_{falsealarm}$ | 0 | 0 | 0 |
| 160 | $P_{detect}$ | 1 | 1 | 1 |
| | $P_{falsealarm}$ | 0 | 0 | 0 |
| 224 | $P_{detect}$ | 1 | 1 | 1 |
| | $P_{falsealarm}$ | 0 | 0 | 0 |
| | # of trials | 1000 | 1000 | 1000 |

## This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

## BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

☐ BLACK BORDERS

☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES

☐ FADED TEXT OR DRAWING

☑ BLURRED OR ILLEGIBLE TEXT OR DRAWING

☐ SKEWED/SLANTED IMAGES

☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS

☐ GRAY SCALE DOCUMENTS

☑ LINES OR MARKS ON ORIGINAL DOCUMENT

☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY

☐ OTHER: ______________________________

**IMAGES ARE BEST AVAILABLE COPY.**
**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**

✓

# PayWord and MicroMint: Two simple micropayment schemes

Ronald L. Rivest* and Adi Shamir**

May 7, 1996

*MIT Laboratory for Computer Science
545 Technology Square, Cambridge, Mass. 02139

**Weizmann Institute of Science
Applied Mathematics Department
Rehovot, Israel

{rivest,shamir}@theory.lcs.mit.edu

# 1 Introduction

We present two simple micropayment schemes, "PayWord" and "MicroMint," for making small purchases over the Internet. We were inspired to work on this problem by DEC's "Millicent" scheme[10]. Surveys of some electronic payment schemes can be found in Hallam-Baker [6], Schneier[16], and Wayner[18].

Our main goal is to minimize the number of public-key operations required per payment, using hash operations instead whenever possible. As a rough guide, hash functions are about 100 times faster than RSA signature verification, and about 10,000 times faster than RSA signature generation: on a typical workstation, one can sign two messages per second, verify 200 signatures per second, and compute 20,000 hash function values per second.

To support micropayments, exceptional efficiency is required, otherwise the cost of the mechanism will exceed the value of the payments. As a consequence, our micropayment schemes are light-weight compared to full macropayment schemes. We "don't sweat the small stuff": a user who loses a micropayment is similar to someone who loses a nickel in a candy machine. Similarly, candy machines aren't built with expensive mechanisms for detecting forged coins, and yet they work well in practice, and the overall level of abuse is low. Large-scale and/or persistent fraud must be detected and eliminated, but if the scheme delivers a volume of payments to the right parties that is roughly correct, we're happy.

In our schemes the players are brokers, users, and vendors. Brokers authorize users to make micropayments to vendors, and redeem the payments collected by the vendors. While user-vendor relationships are transient, broker-user and broker-vendor relationships are long-term. In a typical transaction a vendor sells access to a World-Wide Web page for one cent. Since a user may access only a few pages before moving on, standard credit-card arrangements incur unacceptably high overheads.

The first scheme, "PayWord," is a credit-based scheme, based on chains of "paywords" (hash values). Similar chains have been previously proposed for different purposes: by Lamport [9] and Haller (in S/Key) for access control [7], and by Winternitz [11] as a one-time signature scheme. The application of this idea for micropayments has also been independently discovered by Anderson et al. [2] and by Pederson [14], as we learned after distributing the initial draft of this paper. We discuss these related proposals further in Section 5. The user authenticates a complete chain to the vendor with a single public-key signature, and then successively reveals each payword in the chain to the vendor to make micropayments. The incremental cost of a payment is thus one hash function computation per party. PayWord is optimized for sequences of micropayments, but is secure and flexible enough to support larger variable-value payments as well.

The second scheme, "MicroMint," was designed to eliminate public-key operations altogether. It has lower security but higher speed. It introduces a new paradigm of representing coins by $k$-way hash-function collisions. Just as for a real mint, a broker's "economy of scale" allows him to produce large quantities of such coins at very low cost per coin, while small-scale forgery attempts can only produce coins at a cost exceeding their value.

## 2 Generalities and Notation

We use public-key cryptography (e.g. RSA with a short public exponent). The public keys of the broker $B$, user $U$, and vendor $V$ are denoted $PK_B$, $PK_U$, and $PK_V$, respectively; their secret keys are denoted $SK_B$, $SK_U$, and $SK_V$. A message $M$ with its digital signature produced by secret key $SK$ is denoted $\{M\}_{SK}$. This signature can be verified using the corresponding public key $PK$.

We let $h$ denote a cryptographically strong hash function, such as MD5[15] or SHA[13]. The output (nominally 128 or 160 bits) may be truncated to shorter lengths as described later. The important property of $h$ is its one-wayness and collision-resistance; a very large search should be required to find a single input producing a given output, or to find two inputs producing the same output. The input length may, in some cases, be equal to the output length.

## 3 PayWord

PayWord is credit-based. The user establishes an account with a broker, who issues her a digitally-signed PayWord Certificate containing the broker's name, the user's name and IP-address, the user's public key, the expiration date, and other information. The certificate has to be renewed by the broker (e.g. monthly), who will do so if the user's account is in good standing. This certificate authorizes the user to make Payword chains, and assures vendors that the user's paywords are redeemable by the broker. We assume in this paper that each payword is worth exactly one cent (this could be varied).

In our typical application, when $U$ clicks on a link to a vendor $V$'s non-free web page, his browser determines whether this is the first request to $V$ that day. For a first request, $U$ computes and signs a "commitment" to a new user-specific and vendor-specific chain of paywords $w_1$, $w_2$, …, $w_n$. The user creates the payword chain in reverse order by picking the last payword $w_n$ at random, and then computing

$$w_i = h(w_{i+1})$$

for $i = n-1, n-2, \ldots, 0$. Here $w_0$ is the root of the payword chain, and is not a payword itself. The commitment contains the root $w_0$, but not any payword $w_i$ for $i > 0$. Then $U$ provides this commitment and her certificate to $V$, who verifies their signatures.

The $i$-th payment (for $i = 1, 2, \ldots$) from $U$ to $V$ consists of the pair $(w_i, i)$, which the vendor can verify using $w_{i-1}$. Each such payment requires no calculations by $U$, and only a single hash operation by $V$.

At the end of each day, $V$ reports to $B$ the last (highest-indexed) payment $(w_l, l)$ received from each user that day, together with each corresponding commitment. $B$ charges $U$'s account $l$ cents and pays $l$ cents into $V$'s account. (The broker might also charge subscription and/or transaction fees, which we ignore here.)

A fundamental design goal of PayWord is to minimize communication (particularly on-line communication) with the broker. We imagine that there will be only a few nationwide brokers; to prevent them from becoming a bottleneck, it is important that their computational burden be both reasonable and "off-line." PayWord is an "off-line" scheme: $V$ does not need to interact with $B$ when $U$ first contacts $V$, nor does $V$ need to interact with $B$ as each payment is made. Note that $B$ does not even receive every payword spent, but only the *last* payword spent by each user each day at each vendor.

PayWord is thus extremely efficient when a user makes repeated requests from the same vendor, but is quite effective in any case. The public-key operations required by $V$ are only signature verifications, which are relatively efficient. We note that Shamir's probabilistic signature screening techniques[17] can be used here to reduce the computational load on the vendor even further. Another application where PayWord is well-suited is the purchase of pay-per-view movies; the user can pay a few cents for each minute of viewing time.

This completes our overview; we now give some technical details.

## 3.1 User-Broker relationship and certificates

User $U$ begins a relationship with broker $B$ by requesting an account and a PayWord Certificate. She gives $B$ over a secure authenticated channel: her credit-card number, her public key $PK_U$, and her "delivery address" $A_U$. Her aggregated PayWord charges will be charged to her credit-card account. Her delivery address is her Internet/email or her U.S. mail address; her certificate will only authorize payments by $U$ for purchases to be delivered to $A_U$.

The user's certificate has an expiration date $E$. Certificates might expire monthly, for example. Users who don't pay their bills won't be issued new certificates.

The broker may also give other (possibly user-specific) information $I_U$ in the certificate, such as: a certificate serial number, credit limits to be applied per vendor, information on how to contact the broker, broker/vendor terms and conditions, etc.

The user's certificate $C_U$ thus has the form:

$$C_U = \{B, U, A_U, PK_U, E, I_U\}_{SK_B} .$$

The PayWord certificate is a statement by $B$ to any vendor that $B$ will redeem authentic paywords produced by $U$ turned in before the given expiration date (plus a day's grace).

PayWord is not intended to provide user anonymity. Although certificates could contain user account numbers instead of user names, the inclusion of $A_U$ effectively destroys $U$'s anonymity. However, some privacy is provided, since there is no record kept as to which documents were purchased.

If $U$ loses her secret key she should report it at once to $B$. Her liability should be limited in such cases, as it is for credit-card loss. However, if she does so repeatedly the broker may refuse her further service. The broker may also keep a "hot list" of certificates whose users have reported lost keys, or which are otherwise problematic.

As an alternative to hot-lists, one can use hash-chains in a different manner as proposed by Micali [12] to provide daily authentication of the user's certificate. The user's certificate would additionally contain the root $w'_0$ of a hash chain of length 31. On day $j-1$ of the month, the broker will send the user (e.g. via email) the value $w'_j$ if and only if the user's account is still in good standing. Vendors will then demand of each user the appropriate $w'$ value before accepting payment.

### 3.2 User-Vendor relationships and payments

User-vendor relationships are transient. A user may visit a web site, purchase ten pages, and then move on elsewhere.

**Commitments**

When $U$ is about to contact a new vendor $V$, she computes a fresh payword chain $w_1$, $\ldots$, $w_n$ with root $w_0$. Here $n$ is chosen at the user's convenience; it could be ten or ten thousand. She then computes her commitment for that chain:

$$M = \{V, C_U, w_0, D, I_M\}_{SK_U} .$$

Here $V$ identifies the vendor, $C_U$ is $U$'s certificate, $w_0$ is the root of the payword chain, $D$ is the current date, and $I_M$ is any additional information that may be desired (such as the length $n$ of the payword chain). $M$ is signed by $U$ and given to $V$. (Since this signature is necessarily "on-line," as it contains the vendor's name, the user might consider using an "on-line/off-line" signature scheme[5].)

This commitment authorizes $B$ to pay $V$ for any of the paywords $w_1$, $\ldots$, $w_n$ that $V$ redeems with $B$ before date $D$ (plus a day's grace). Note that paywords are *vendor-specific* and *user-specific*; they are of no value to another vendor.

Note that $U$ must sign a commitment for each vendor she pays. If she rapidly switches between vendors, the cost of doing so may become noticeable. However, this is PayWord's only significant computational requirement, and the security it provides makes PayWord usable even for larger "macropayments" (e.g. software selling at $19.99).

The vendor verifies $U$'s signature on $M$ and the broker's signature on $C_U$ (contained within $M$), and checks expiration dates.

The vendor $V$ should cache verified commitments until they expire at the end of the day. Otherwise, if he redeemed (and forgot) paywords received before the expiration date of the commitment, $U$ could cheat $V$ by replaying earlier commitments and paywords. (Actually, to defeat this attack, $V$ need store only a short hash of each commitment he has reported to $B$ already today.)

The user should preferably also cache her commitment until she believes that she is finished ordering information from $V$, or until the commitment expires. She can always generate a fresh commitment if she re-visits a vendor whose commitment she has deleted.

**Payments**

The user and vendor need to agree on the amount to be paid. In our exemplary application, the price of a web page is typically one cent, but could be some other amount. A web page should presumably be free if the user has already purchased it that day, and is just requesting it again because it was flushed from his cache of pages.

A payment $P$ from $U$ to $V$ consists of a payword and its index:

$$P = (w_i, i) .$$

The payment is short: only twenty or thirty bytes long. (The first payment to $V$ that day would normally accompany $U$'s corresponding commitment; later payments are just the payword and its index, unless the previous chain is exhausted and a new chain must be committed to.) The payment is not signed by $U$, since it is self-authenticating (using the commitment).

The user spends her paywords in order: $w_1$ first, then $w_2$, and so on. If each payword is worth one cent, and each web page costs one cent, then she discloses $w_i$ to $V$ when she orders her $i$-th web page from $V$ that day.

This leads to the PayWord payment policy: *for each commitment a vendor $V$ is paid $l$ cents, where $(w_l, l)$ is the corresponding payment received with the largest index.* This means that $V$ needs to store only one payment from each user: the one with the highest index. Once a user spends $w_i$, she can not spend $w_j$ for $j < i$. The broker can confirm the value to be paid for $w_l$ by determining how many applications of $h$ are required to map $w_l$ into $w_0$.

*PayWord supports variable-size payments in a simple and natural manner.* If $U$ skips paywords, and gives $w_7$ after giving $w_2$, she is giving $V$ a nickel instead of a penny. When $U$ skips paywords, during verification $V$ need only apply $h$ a number of times proportional to the value of the payment made.

A payment does not specify what item it is payment for. The vendor may cheat $U$ by sending him nothing, or the wrong item, in return. The user bears the risk of losing the payment, just as if he had put a penny in the mail. Vendors who so cheat their customers will be shunned. This risk can be moved to $V$, if $V$ specifies payment *after* the document has been delivered. If $U$ doesn't pay, $V$ can notify $B$ and/or refuse $U$ further service. For micropayments, users and vendors might find either approach workable.

## 3.3 Vendor-Broker relationships and redemption

A vendor $V$ needn't have a prior relationship with $B$, but does need to obtain $PK_B$ in an authenticated manner, so he can authenticate certificates signed by $B$. He also needs to establish a way for $B$ to pay $V$ for paywords redeemed. (Brokers pay vendors by means outside the PayWord system.)

At the end of each day (or other suitable period), $V$ sends $B$ a redemption message giving, for each of $B$'s users who have paid $V$ that day (1) the commitment $C_U$ received from $U$, (2) the last payment $P = (w_l, l)$ received from $U$.

The broker then needs to (1) verify each commitment received (he only needs to verify user signatures, since he can recognize his own certificates), including checking of dates, etc., and (2) verify each payment $(w_l, l)$ (this requires $l$ hash function applications). We assume that $B$ normally honors all valid redemption requests.

Since hash function computations are cheap, and signature verifications are only moderately expensive, $B$'s computational burden should be reasonable, particularly since it is more-or-less proportional to the payment volume he is supporting; $B$ can charge transaction or subscription fees adequate to cover his computation costs. We also note that $B$ never needs to respond in real-time; he can batch up his computations and perform them off-line overnight.

### 3.4 Efficiency

We summarize PayWord's computational and storage requirements:

- The broker needs to sign each user certificate, verify each user commitment, and perform one hash function application per payment. (All these computations are off-line.) The broker stores copies of user certificates and maintains accounts for users and vendors.
- The user needs to verify his certificates, sign each of his commitments, and perform one hash function application per payword committed to. (Only signing commitments is an on-line computation.) He needs to store his secret key $SK_U$, his active commitments, the corresponding payword chains, and his current position in each chain.
- The vendor verifies all certificates and commitments received, and performs one hash function application per payword received or skipped over. (All his computations are on-line.) The vendor needs to store all commitments and the last payment received per commitment each day.

### 3.5 Variations and Extensions

In one variation, $h(\cdot)$ is replaced by $h_s(\cdot) = h(s, \cdot)$, where $s$ is a "salt" (random value) specified in the commitment. Salting may enable the use of faster hash functions or hash functions with a shorter output length (perhaps as short as 64–80 bits).

The value of each payword might be fixed at one cent, or might be specified in $C_U$ or $M$. In a variation, $M$ might authenticate several chains, whose paywords have different values (for penny paywords, nickel paywords, etc.).

The user name may also need to be specified in a payment if it is not clear from context. If $U$ has more than one payword chain authorized for $V$, then the payment should specify which is relevant.

Paywords could be sold on a debit basis, rather than a credit basis, but only if the user interacts with the broker to produce each commitment: the certificate could require that the broker, rather than the user, sign each commitment. The broker can automatically refund the user for unused paywords, once the vendor has redeemed the paywords given to him.

In some cases, for macropayments, it might be useful to have the "commitment" act like an electronic credit card order or check without paywords being used at all. The commitment would specify the vendor and the amount to be paid.

The broker may specify in user certificates other terms and conditions to limit his risk. For example, $B$ may limit the amount that $U$ can spend per day at any vendor. Or, $B$ may refuse payment if $U$'s name is on $B$'s "hot list" at the beginning of the day. (Vendors can down-load $B$'s hot-list each morning.) Or, $B$ may refuse to pay if $U$'s total expenditures over all vendors exceeds a specified limit per day. This protects $B$ from extensive liability if $SK_U$ is stolen and abused. (Although again, since $C_U$ only authorizes delivery to $A_U$, risk is reduced.) In these cases vendors share the risk with $B$.

Instead of using payword chains, another method we considered for improving efficiency was to have $V$ *probabilistically* select payments for redemption. We couldn't make this idea work out, and leave this approach as an open problem.

## 4 MicroMint

MicroMint is designed to provide reasonable security at very low cost, and is optimized for unrelated low-value payments. MicroMint uses *no* public-key operations at all.

MicroMint "coins" are produced by a broker, who sells them to users. Users give these coins to vendors as payments. Vendors return coins to the broker in return for payment by other means.

A coin is a bit-string whose validity can be easily checked by anyone, but which is hard to produce. This is similar to the requirements for a public-key signature, whose complexity makes it an overkill for a transaction whose value is one cent. (PayWord uses signatures, but not on every transaction.)

MicroMint has the property that generating many coins is very much cheaper, per coin generated, than generating few coins. A large initial investment is required to generate the first coin, but then generating additional coins can be made progressively cheaper. This is similar to the economics for a regular mint, which invests in a lot of expensive machinery to make coins economically. (It makes no sense for a forger to produce coins in a way that costs more per coin produced than its value.)

The broker will typically issue new coins at the beginning of each month; the validity of these coins will expire at the end of the month. Unused coins are returned to the broker at the end of each month, and new coins can be purchased at the beginning of each month. Vendors can return the coins they collect to the broker at their convenience (e.g. at the end of each day).

We now describe the "basic" variant of MicroMint. Many extensions and variations are possible on this theme; we describe some of them in section 4.2.

**Hash Function Collisions**

MicroMint coins are represented by *hash function collisions*, for some specified one-way hash function $h$ mapping $m$-bit strings $x$ to $n$-bit strings $y$. We say that $x$ is a pre-image of $y$ if $h(x) = y$. A pair of distinct $m$-bit strings $(x_1, x_2)$ is called a *(2-way) collision* if $h(x_1) = h(x_2) = y$, for some $n$-bit string $y$.

If $h$ acts "randomly," the only way to produce even one acceptable 2-way collision is to hash about $\sqrt{2^n} = 2^{n/2}$ $x$-values and search for repeated outputs. This is essentially the "birthday paradox." (We ignore small constants in our analyses.)

Hashing $c$ times as many $x$-values as are needed to produce the first collision results in approximately $c^2$ as many collisions, for $1 \leq c \leq 2^{n/2}$, so producing collisions can be done increasingly efficiently, per coin generated, once the threshold for finding collisions has been passed.

**Coins as $k$-way collisions**

A problem with 2-way collisions is that choosing a value of $n$ small enough to make the broker's work feasible results in a situation where coins can be forged a bit too easily by an adversary. To raise the threshold further against would-be forgers, we propose using $k$-way collisions instead of 2-way collisions.

A $k$-way collision is a set of $k$ distinct $x$-values $x_1, x_2, \ldots, x_k$ that have the same hash value $y$. The number of $x$-values that must be examined before one expects to see the first $k$-way collision is then approximately $2^{n(k-1)/k}$. If one examines c times this many $x$-values, for $1 \leq c \leq 2^{n/k}$, one expects to see about $c^k$ $k$-way collisions. Choosing $k > 2$ has the dual effect of delaying the threshold where the first collision is seen, and also accelerating the rate of collision generation, once the threshold is passed.

We thus let a $k$-way collision $(x_1, \ldots, x_k)$ represent a coin. The validity of this coin can be easily verified by anyone by checking that the $x_i$'s are distinct and that

$$h(x_1) = h(x_2) = \cdots = h(x_k) = y$$

for some $n$-string $y$.

### Minting coins

The process of computing $h(x) = y$ is analogous to tossing a ball ($x$) at random into one of $2^n$ bins; the bin that ball $x$ ends up in is the one with index $y$. A coin is thus a set of $k$ balls that have been tossed into the same bin. Getting $k$ balls into the same bin requires tossing a substantial number of balls altogether, since balls can not be "aimed" at a particular bin. To mint coins, the broker will create $2^n$ bins, toss approximately $k2^n$ balls, and create one coin from each bin that now contains at least $k$ balls. With this choice of parameters each ball has a chance of roughly 1/2 of being part of a coin.

Whenever one of the $2^n$ bins has $k$ or more balls in it, $k$ of those balls can be extracted to form a coin. Note that if a bin has more than $k$ balls in it, the broker can in principle extract $k$-subsets in multiple ways to produce several coins. However, an adversary who obtains two different coins from the same bin could combine them to produce multiple new coins. Therefore, we recommend that *a MicroMint broker should produce at most one coin from each bin.* Following this rule also simplifies the Broker's task of detecting multiply-spent coins, since he needs to allocate a table of only $2^n$ bits to indicate whether a coin with a particular $n$-bit hash value has already been redeemed.

A small problem in this basic picture, however, is that computation is much cheaper than storage. The number of balls that can be tossed into bins in a month-long computation far exceeds both the number of balls that can be memorized on a reasonable number of hard disks and the number of coins that the broker might realistically need to mint. One could attempt to balance the computation and memory requirements by utilizing a very slow hash algorithm, such as DES iterated many times. Unfortunately, this approach also slows down the verification process.

A better approach, which we adopt, is to make most balls unusable for the purpose of minting coins. To do so, we say that a ball is "good" if the high-order bits of the hash value $y$ have a value $z$ specified by the broker. More precisely, let $n = t + u$ for some specified nonnegative integers $t$ and $u$. If the high-order $t$ bits of $y$ are equal to the specified value $z$ then the value $y$ is called "good, " and the low-order $u$ bits of $y$ determine the index of the bin into which the (good) ball $x$ is tossed. (General $x$ values are referred to merely as "balls," and those that are not good can be thought of as having been conceptually tossed into nonexistent virtual bins that are "out of range.")

A proper choice of $t$ enables us to balance the computational and storage requirements of the broker, without slowing down the verification process. It slows down the generation process by a factor of $2^t$, while limiting the storage requirements of the broker to a small multiple of the number of coins to be generated. The broker thus tosses approximately $k2^u$ balls, memorizes about $k2^u$ good balls that he tosses into the $2^u$ bins, and generates from them approximately $(1/2) \cdot 2^u$ valid coins.

Remark: We note that with standard hash functions, such as MD5 and DES, the number of ouput bits produced may exceed the number $n$ of bits specified in the broker's parameters. A suitable hash function for the broker can be obtained by discarding all but the low-order $n$ bits of the standard hash function output. This discarding of bits other than the low-order $n$ bits is a different process than that of specifying a particular value for the high-order $t$ bits out of the $n$ that was described above.

## A detailed scenario

Here is a detailed sketch of how a typical broker might proceed to choose parameters for his minting operating for a given month. The calculations are approximate (values are typically rounded to the nearest power of two), but instructive; they can be easily modified for other assumptions.

The broker will invest in substantial hardware that gives him a computational advantage over would-be forgers, and run this hardware continuously for a month to compute coins valid for the next month. This hardware is likely to include many special-purpose chips for computing $h$ efficiently.

We suppose that the broker wishes to have a net profit of $1 million per month (approximately $2^{27}$ cents/month). He charges a brokerage fee of 10%. That is, for every coin worth one cent that he sells, he only gives the vendor 0.9 cents when it is redeemed. Thus, the broker needs to sell one billion coins per month (approximately $2^{30}$ coins/month) to collect his $1M fee. If an average user buys 2500 ($25.00) coins per month, he will need to have a customer base of 500,000 customers.

The broker chooses $k = 4$; a coin will be a good 4-way collision.

To create $2^{30}$ coins, the broker chooses $u = 31$, so that he creates an array of $2^{31}$ (approximately two billion) bins, each of which can hold up to 4 $x$-values that hash to an $n$-bit value that is the concantenation of a fixed $t$-bit pattern $z$ and the $u$-bit index of the bin.

The broker will toss an average of 4 balls into each bin. That is, the broker will generate $4 \cdot 2^{31} = 2^{33}$ (approximately eight billion) $x$-values that produce good $y$-values. When he does so, the probability that a bin then contains 4 or more $x$-values (and thus can yield a coin) is about 1/2. (Using a Poisson approximation, it can be calculated that the correct value is approximately 0.56.) Since each of the $2^{31}$ bins produces a coin with probability 1/2, the number of coins produced is $2^{30}$, as desired.

In order to maximize his advantage over an adversary who wishes to forge coins, the broker invests in special-purpose hardware that allows him to compute hash values very quickly. This will allow him to choose a relatively large value of $t$, so that good hash values are relatively rare. This increases the work factor for an adversary (and for the broker) by a factor of $2^t$. The broker chooses his hash function $h$ as the low-order $n$ bits of the encryption of some fixed value $v_0$ with key $x$ under the Data Encryption Standard (DES):

$$h(x) = [DES_x(v_0)]_{1\ldots n} .$$

The broker purchases a number of field-programmable gate array (FPGA) chips, each of which is capable of hashing approximately $2^{25}$ (approximately 30 million) $x$-values per second. (See [3].) Each such chip costs about \$200; we estimate that the broker's actual cost per chip might be closer to \$400 per chip when engineering, support, and associated hardware are also considered. The broker purchases $2^8$ (= 256) of these chips, which costs him about \$100,000. These chips can collectively hash $2^{33}$ (approximately 8.6 billion) values per second. Since there are roughly $2^{21}$ (two million) seconds in a month, they can hash about $2^{54}$ (approximately 18 million billion) values per month.

Based on these estimates the broker chooses $n = 52$ and $t = 21$ and runs his minting operation for one month. Of the $k2^n = 2^{54}$ hash values computed, only one in $2^{21}$ will be good, so that approximately $2^{33}$ good $x$-values are found, as necessary to produce $2^{30}$ coins.

Storing a good $(x, h(x))$ pair takes less than 16 bytes. The total storage required for all good pairs is less than $2^{37}$ bytes (128 Gigabytes). Using standard magnetic hard disk technology costing approximately \$300 per Gigabyte, the total cost for storage is less than \$40,000. The total cost for the broker's hardware is thus less than \$150,000, which is less than 15% of the first month's profit.

Rather than actually writing each pair into a randomly-accessible bin, the broker can write the $2^{33}$ good pairs sequentially to the disk array, and then sort them into increasing order by $y$ value, to determine which are in the same bin. With a reasonable sorting algorithm, the sorting time should be under one day.

### Selling coins

Towards the end of each month, the broker begins selling coins to users for the next month. At the beginning of each month, $B$ reveals the new validity criterion for coins to be used that month. Such sales can either be on a debit basis or a credit basis, since $B$ will be able to recognize coins when they are returned to him for redemption. In a typical purchase, a user might buy \$25.00 worth of coins (2500 coins), and charge the purchase to his credit card. The broker keeps a record of which coins each user bought. Unused coins are returned to the broker at the end of each month.

### Making payments

Each time a user purchases a web page, he gives the vendor a previously unspent coin $(x_1, x_2, \ldots, x_k)$. (This might be handled automatically by the user's web browser when the user clicks on a link that has a declared fee.) The vendor verifies that it is indeed a good $k$-way collision by computing $h(x_i)$ for $1 \leq i \leq k$, and checking that the values are equal and good. Note that while the broker's minting process was intentionally slowed down by a factor of $2^t$, the vendor's task of verifying a coin remains extremely efficient, requiring only $k$ hash computations and a few comparisons (in our proposed scenario, $k = 4$).

### Redemptions

The vendor returns the coins he has collected to the broker at the end of each day. The broker checks each coin to see if it has been previously returned, and if not, pays the vendor one cent (minus his brokerage fee) for each coin. We propose that if the broker receives a specific coin more than once, he does not pay more than once. Which vendor gets paid can be decided arbitrarily or randomly by the broker. This may penalize vendors, but eliminates any financial motivation a vendor might have had to cheat by redistributing coins he has collected to other vendors.

## 4.1 Security Properties

We distinguish between small-scale attacks and large-scale attacks. We believe that users and vendors will have little motivation to cheat in order to gain only a few cents; even if they do, the consequences are of no great concern. This is similar to the way ordinary change is handled: many people don't even bother to count their change following a purchase. Our security mechanisms are thus primarily designed to discourage large-scale attacks, such as massive forgery or persistent double-spending.

**Forgery**

Small-scale forgery is too expensive to be of interest to an adversary: with the recommended choice of $k = 4$, $n = 54$, and $u = 31$, the generation of the first forged coin requires about $2^{45}$ hash operations. Since a standard work-station can perform only $2^{14}$ hash operations per second, a typical user will need $2^{31}$ seconds (about 80 years) to generate just one forged coin on his workstation.

Large-scale forgery can be detected and countered as follows:

- All forged coins automatically become invalid at the end of the month.
- Forged coins can not be generated until after the broker announces the new monthly coin validity criterion at the beginning of the month.
- The use of hidden predicates (described below) gives a finer time resolution for rejecting forged coins without affecting the validity of legal coins already in circulation.
- The broker can detect the presence of a forger by noting when he receives coins correspondings to bins that he did not produce coins from. This works well in our scenario since only about half of the bins produce coins. To implement this the broker need only work with a bit-array having one bit per bin.
- The broker can at any time declare the current period to be over, recall all coins for the current period, and issue new coins using a new validation procedure.
- The broker can simultaneously generate coins for several future months in a longer computation, as described below; this makes it harder for a forger to catch up with the broker.

**Theft of coins**

If theft of coins is judged to be a problem during initial distribution to users or during redemption by vendors, it is easy to transmit coins in encrypted form during these operations.

User/broker and vendor/broker relationships are relatively stable, and long-term encryption keys can be arranged between them.

To protect coins as they are being transferred over the Internet from user to vendor, one can of course use public-key techniques to provide secure communication. However, in keeping with our desire to minimize or eliminate public-key operations, we propose below another mechanism, which makes coins user-specific. This does not require public-key cryptography, and makes it harder to re-use stolen coins.

Another concern is that two vendors may collude so that both attempt to redeem the same coins. The recommended solution is that a broker redeem a coin at most once, as discussed earlier. Since this may penalize honest vendors who receive stolen coins, we can make coins vendor-specific as well as user-specific, as described below.

#### Double-spending

Since the MicroMint scheme is not anonymous, the broker can detect a doubly-spent coin, and can identify which vendors he received the two instances from. He also knows which user the coin was issued to. With the vendors' honest cooperation, he can also identify which users spent each instance of that coin. Based on all this information, the broker can keep track of how many doubly-spent coins are asssociated with each user and vendor. A large-scale cheater (either user or vendor) can be identified by the large number of duplicate coins associated with his purchases or redemptions; the broker can then drop a large-scale cheater from the system. A small-scale cheater may be hard to identify, but, due to the low value of individual coins, it is not so important if he escapes identification.

MicroMint does not provide any mechanism for preventing purely malicious framing (with no financial benefit to the framer). We believe that the known mechanisms for protecting against such behavior are too cumbersome for a light-weight micropayment scheme. Since MicroMint does not use real digital signatures, it may be hard to legally prove who is guilty of duplicating coins. Thus, a broker will not be able to pursue a cheater in court, but can always drop a suspected cheater from the system.

### 4.2 Variations

#### User-specific coins

We describe two proposals for making coins that are user-specific in a way that can be easily checked by vendors. Such coins, if stolen, are of no value to most other users. This greatly reduces the motivation for theft of coins.

In the first proposal, the broker splits the users into "groups," and gives each user coins whose validity depends on the identity of the group. For example, the broker can give user $U$ coins that satisfy the additional condition $h'(x_1, x_2, \ldots, x_k) = h'(U)$, where hash function $h'$ produces short (e.g. 16-bit) output values that indicate $U$'s group. A vendor can easily check this condition, and reject a coin that is not tendered by a member of the correct group.

The problem with this approach is that if the groups are too large, then a thief can easily find users of the appropriate group who might be willing to buy stolen coins. On the other hand, if the groups are too small (e.g. by placing each user is in his own group), the broker may be forced to precompute a large excess of coins, just to ensure that he has a large enough supply to satisfy each user's unpredictable needs.

In the second proposal, we generalize the notion of a "collision" to more complicated combinatorial structures. Formally, a coin $(x_1, \ldots, x_k)$ will be valid for a user $U$ if the images $y_1 = h(x_1)$, $y_2 = h(x_2)$, $\ldots$, $y_k = h(x_k)$ satisfy the condition

$$y_{i+1} - y_i = d_i \pmod{2^u}$$

for $i = 1, 2, \ldots, k-1$, where

$$(d_1, d_2, \ldots, d_{k-1}) = h'(U)$$

for a suitable auxiliary hash function $h'$. (The original proposal for representing coins as collisions can be viewed as the special case where all the distances $d_i$'s between the $k$ bins are zero.)

To mint coins of this form, the broker fills up most of his bins by randomly tossing balls into them, except that now it is not necessary to have more than one ball per bin. We emphasize that this pre-computation is not user-specific, and the broker does not need to have any prior knowledge of the number of coins that will be requested by each user, since each good ball can be used in a coin for *any* user. After this lengthy pre-computation, the broker can quickly create a coin for any user $U$ by

- Computing $(d_1, \ldots, d_{k-1}) = h'(U)$.
- Picking a random bin index $y_1$. (This bin should have been previously unused as a $y_1$ for another coin, so that $y_1$ can be used as the "identity" of the coin when the broker uses a bit-array to determine which coins have already been redeemed.)
- Computing $y_{i+1} = y_i + d_i \pmod{2^u}$ for $i = 1, 2, \ldots, k-1$,
- Taking a ball $x_1$ out of bin $y_1$, and taking a copy of one ball out of each bin $y_2, \ldots, y_k$. (If any bin $y_i$ is empty, start over with a new $y_1$.) Note that balls may be re-used in this scheme.
- Producing the ordered $k$-tuple $(x_1, \ldots, x_k)$ as the output coin.

A convenient feature of this scheme is that it is easy to produce a large number of coins for a given user even when the broker's storage device is a magnetic disk with a relatively slow seek time. The idea is based on the observation that if the $y_1$ values for successive coins are consecutive, then so also will be the $y_i$ values for each $i$, $1 < i \leq k$. Therefore, a request for 2500 new coins with $k = 4$ requires only four disk seeks, rather than 10,000 seeks: at 10 milliseconds per seek, this reduces the total seek time from 100 seconds to only 40 milliseconds.

Note that in principle coins produced for different users could re-use the same ball $x_i$. Conceivably, someone could forge a new coin by combining pieces of other coins he has seen. However, he is unlikely to achieve much success by this route unless he sees balls from a significant fraction of all the bins. For example, suppose that there are $2^{31}$ bins, of which the forger has seen a fraction $2^{-10}$ (i.e., he has collected $2^{21}$ balls from coins spent by other users). Then the expected number of coins he can piece together from these balls that satisfy the condition of being a good coin for himself is only $2^{31}(2^{-10})^3 = 2$. (Even if he had 1000 customers for these coins, he would expect to make only 2000 coins total, or two coins per customer on the average.) Thus, we are not too concerned about this sort of "cut-and-paste" forgery.

**Vendor-specific coins**

To further reduce the likelihood that coins will be stolen, the user can give coins to vendors in such a way that each coin can be redeemed only by a small fraction of the vendors. This technique makes a stolen coin less desirable, since it is unlikely to be accepted by a vendor other than the one where it was originally spent. The additional check of validity can be carried out both by the vendor and by the broker. (Having vendor-specific coins is also a major feature of the Millicent [10] scheme.)

The obvious difficulty is that neither the broker nor the user can predict ahead of time which vendors the user will patronize, and it is unreasonable to force the user to purchase in advance coins specific for each possible vendor. Millicent adopts the alternative strategy whereby the user must contact the broker in real-time whenever the user needs coins for a new vendor. (He also needs to contact the broker to return excess unused coins that are specific to that vendor.) We can overcome these problems with an extension of the user-specific scheme described above, in which the user purchases a block of "successive" MicroMint coins.

Intuitively, the idea is the following. Choose a value $v$ (e.g. 1024) less than $u$. Let a $u$-bit bin-index $y$ be divided into a $u-v$-bit upper part $y'$ and a $v$-bit lower part $y''$. We consider that $y'$ specifies a "superbin" index and that $y''$ specifies a bin within that superbin. A user now purchases balls in bulk and makes his own coins. He purchases balls by the superbin, obtaining $2^v$ balls per superbin with one ball in each bin of the superbin. He buy $k$ superbins of balls for $2^v$ cents. A coin from user $U$ is valid for redemption by vendor $V$ if:

$$y'_{i+1} = y'_i + d'_i \pmod{2^{u-v}} \text{ for } i = 1, \ldots, k-1,$$

and

$$y''_{i+1} = y''_i + d''_i \pmod{2^v} \text{ for } i = 1, \ldots, k-1,$$

where

$$h'(U) = (d'_1, \ldots, d'_{k-1})$$

and

$$h''(V) = (d''_1, \ldots, d''_{k-1}) .$$

The broker chooses the next available superbin as the first superbin to give the user; the other superbins are then uniquely determined by the differences $\{d'_i\}$ defined by the user's identity and the choice of the first superbin. Analogously, to make a coin for a particular vendor the user chooses a ball from the next bin from his first superbin, and must use balls from bins in the other superbins that are then uniquely determined by the differences $\{d''_i\}$ defined by the vendor's identity and the choice of the first bin. Note that balls from the first superbin are used only once, to permit detection of double-spending, whereas balls from the other superbins may appear more than once (in coins paid to different vendors), or not at all. It may be difficult for a broker to create superbins that are perfectly full even if he throws more balls. He might sell superbins that are almost full, but then a user may have difficulty producing some coins for some vendors. To compensate, the broker can reduce the price by one cent for each empty bin sold.

### Simultaneously generating balls for multiple months

Our major line of defense against large-scale forgery is the fact that the broker can compute coins in advance, whereas a forgery attempt can only be started once the new validity condition for the current month is announced. We now describe a technique whereby computing the balls for a single month's coins takes eight months, but the broker doesn't fall behind because he can generate balls for eight future months concurrently. The forger will thus have the dual problems of starting late and being too slow, even if he uses the same computational resources as the real broker.

In this method, the broker changes the monthly validity criterion, not by changing the hash function $h$, but by announcing each month a new value $z$ such that ball $x$ is good when the high-order $t$ bits of $h(x)$ are equal to $z$. The broker randomly and secretly chooses in advance the values $z$ that will be used for each of the next eight months. Tossing a ball still means performing one hash function computation, but the tossed ball is potentially "good" for any of the next eight months, and it is trivial for the broker to determine if this is the case. In contrast, the forger only knows the current value of $z$, and can not afford to memorize all the balls he tosses, since memory is relatively expensive and only a tiny fraction (e.g., $2^{-21}$ in our running example) of the balls are considered "good" at any given month.

We now describe a convenient way of carrying out this calculation. Assume that at the beginning of the month $j$, the broker has all of the balls needed for month $j$, 7/8 of the balls needed for month $j+1$, 6/8 of the balls needed for month $j+2$, ..., and 1/8 of the balls needed in for month $j+7$. During month $j$, the broker tosses balls by randomly picking $x$ values, calculating $y = h(x)$, and checking whether the top-most $t$ bits of $y$ are equal to any of the $z$ values to be used in months $j+1, \ldots, j+8$. To slow the rate at which he generates good balls for each upcoming month, he increases $n$ and $t$ each by three. After the month-long computation, we expect him to have all the coins he needs for month $j+1$, 7/8 of the coins he needs for month $j+2$, and so on; this is the desired "steady-state" situation. The broker needs four times as much storage to hold the balls generated for future months, but balls for future months can be temporarily stored on inexpensive magnetic tapes because he doesn't need to respond quickly to user requests for those coins yet.

### Hidden Predicates

The "hidden predicate" technique for defeating forgers works as follows. We choose $m > n$, and require each $m$-bit pre-image to satisfy a number of hidden predicates. The hidden predicates should be such that generating pre-images satisfying the predicates is easy (if you know the predicate). To generate an $x_i$, one can pick its last $n$ bits randomly, and define the $j$-th bit of $x_i$, for $j = m-n, \ldots, 1$, to be the $j$-th hidden predicate applied to bits $j+1, \ldots, m$ of $x_i$. The hidden predicates must be balanced and difficult to learn from random examples. Suggestions of hard-to-learn predicates exist in the learning-theory literature. For example the parity/majority functions of Blum et al.[4] (which are the exclusive-or of some of the input bits together with the majority function on a disjoint set of input bits) are interesting, although slightly more complicated functions may be appropriate in this application when word lengths are short. With $m - n = 32$, the broker can have one hidden predicate for each day of the month. He could reveal a new predicate each day, and ask vendors to check that the coins they receive satisfy these predicates (otherwise the coins will not be accepted by the broker). This would not affect the validity of legitimate coins already in circulation, but makes forgery extremely difficult, since the would-be forger would have to discard much of his precomputation work as each new predicate is revealed. We feel that such techniques are strongly advisable in MicroMint.

**Other Extensions**

Peter Wayner (private communication) has suggested a variation on MicroMint in which coins of different values are distinguished by publicly-known predicates on the $x$-values.

## 5 Relationship to Other Micropayment Schemes

In this section we compare our proposals to the Millicent[10], NetBill [1], NetCard [2], and Pederson [14] micropayment schemes.

NetBill offers a number of advanced features (such as electronic purchase orders and encryption of purchased information), but it is relative expensive: digital signatures are heavily used and the NetBill server is involved in each payment.

Millicent uses hash functions extensively, but the broker must be on-line whenever the user wishes to interact with a new vendor. The user buys vendor-specific scrip from the broker. For applications such as web browsing, where new user-vendor relationships are continually being created, Millicent can place a heavy real-time burden on the broker. Compared to Millicent, both PayWord and MicroMint enable the user to generate vendor-specific "scrip" without any interaction with the broker, and without the overhead required in returning unused vendor-specific scrip. Also, PayWord is a credit rather than debit scheme.

Anderson, Manifavas, and Sutherland [2] have developed a micropayment system, "NetCard," which is very similar to PayWord in that it uses chains of hash values with a digitally signed root. (The way hash chains are created differs in a minor way.) However, in their proposal, it is the bank rather than the user who prepares the chain and signs the root, which adds to the overall burden of the bank. This approach prevents the user from creating new chains, although a NetCard user could spend a single chain many times. Compared to PayWord, NetCard is debit-based, rather than credit-based. We have heard that a patent has been applied for on the NetCard system.

Torben Pedersen outlines a micropayment proposal[14] that is also based on hash chains. His motivating application was for incremental payment of telephone charges. His paper does not provide much detail on many points (e.g. whether the system is credit or debit-based, how to handle exceptions, whether chains are vendor-specific, and other auxiliary security-related matters). The CAFE project has filed for a patent on what we believe is an elaboration of Pedersen's idea. (The details off the CAFE scheme are not available to us.)

Similarly following Pedersen's exposition, the iKP developers Hauser, Steiner, and Waidner have independently adopted a similar approach [8].

## 6 Conclusions and Discussion

We have presented two new micropayment schemes which are exceptionally economical in terms of the number of public-key operations employed. Furthermore, both schemes are *off-line* from the broker's point of view.

## References

[1] The NetBill Electronic Commerce Project, 1995. `http://www.ini.cmu/NETBILL/home.html`.

[2] Ross Anderson, Harry Manifavas, and Chris Sutherland. A practical electronic cash system, 1995. Available from author: `Ross.Anderson@cl.cam.ac.uk`.

[3] Matt Blaze, Whitfield Diffie, Ronald L. Rivest, Bruce Schneier, Tsutomu Shimomura, Eric Thompson, and Michael Wiener. Minimal key lengths for symmetric ciphers to provide adequate commercial security: A report by an ad hoc group of cryptographers and computer scientists, January 1996. Available at `http://www.bsa.org`.

[4] Avrim Blum, Merrick Furst, Michael Kearns, and Richard J. Lipton. Cryptographic primitives based on hard learning problems. In Douglas R. Stinson, editor, *Proc. CRYPTO 93*, pages 278–291. Springer, 1994. Lecture Notes in Computer Science No. 773.

[5] Shimon Even, Oded Goldreich, and Silvio Micali. On-line/off-line digital signatures. In G. Brassard, editor, *Proc. CRYPTO 89*, pages 263–277. Springer-Verlag, 1990. Lecture Notes in Computer Science No. 435.

[6] Phillip Hallam-Baker. W3C payments resources, 1995. `http://www.w3.org/hypertext/WWW/Payments/overview.html`.

[7] Neil M. Haller. The S/KEY one-time password system. In *ISOC*, 1994.

[8] Ralf Hauser, Michael Steiner, and Michael Waidner. Micro-Payments based on iKP, December 17, 1995. Available from authors. `sti@zurich.ibm.com`.

[9] Leslie Lamport. Password authentication with insecure communication. *Communications of the ACM*, 24(11):770–771, November 1981.

[10] Mark S. Manasse. Millicent (electronic microcommerce), 1995. `http://www.research.digital.com/SRC/personal/Mark_Manasse/uncommon/ucom.html`.

[11] Ralph C. Merkle. A certified digital signature. In G. Brassard, editor, *Proc. CRYPTO 89*, pages 218–238. Springer-Verlag, 1990. Lecture Notes in Computer Science No. 435.

[12] Silvio Micali. Efficient certificate revocation. Technical Report TM-542b, MIT Laboratory for Computer Science, March 22, 1996.

[13] National Institute of Standards and Technology (NIST). *FIPS Publication 180: Secure Hash Standard (SHS)*, May 11, 1993.

[14] Torben P. Pedersen. Electronic payments of small amounts. Technical Report DAIMI PB-495, Aarhus University, Computer Science Department, Århus, Denmark, August 1995.

[15] Ronald L. Rivest. The MD5 message-digest algorithm. Internet Request for Comments, April 1992. RFC 1321.

[16] Bruce Schneier. *Applied Cryptography (Second Edition)*. John Wiley & Sons, 1996.

[17] Adi Shamir. Fast signature screening. CRYPTO '95 rump session talk; to appear in RSA Laboratories' *CryptoBytes*.

[18] Peter Wayner. *Digital Cash: Commerce on the Net.* Academic Press, 1996.

User/broker and vendor/broker relationships are relatively stable, and long-term encryption keys can be arranged between them.

To protect coins as they are being transferred over the Internet from user to vendor, one can of course use public-key techniques to provide secure communication. However, in keeping with our desire to minimize or eliminate public-key operations, we propose below another mechanism, which makes coins user-specific. This does not require public-key cryptography, and makes it harder to re-use stolen coins.

Another concern is that two vendors may collude so that both attempt to redeem the same coins. The recommended solution is that a broker redeem a coin at most once, as discussed earlier. Since this may penalize honest vendors who receive stolen coins, we can make coins vendor-specific as well as user-specific, as described below.

**Double-spending**

Since the MicroMint scheme is not anonymous, the broker can detect a doubly-spent coin, and can identify which vendors he received the two instances from. He also knows which user the coin was issued to. With the vendors' honest cooperation, he can also identify which users spent each instance of that coin. Based on all this information, the broker can keep track of how many doubly-spent coins are asssociated with each user and vendor. A large-scale cheater (either user or vendor) can be identified by the large number of duplicate coins associated with his purchases or redemptions; the broker can then drop a large-scale cheater from the system. A small-scale cheater may be hard to identify, but, due to the low value of individual coins, it is not so important if he escapes identification.

MicroMint does not provide any mechanism for preventing purely malicious framing (with no financial benefit to the framer). We believe that the known mechanisms for protecting against such behavior are too cumbersome for a light-weight micropayment scheme. Since MicroMint does not use real digital signatures, it may be hard to legally prove who is guilty of duplicating coins. Thus, a broker will not be able to pursue a cheater in court, but can always drop a suspected cheater from the system.

## 4.2 Variations

**User-specific coins**

We describe two proposals for making coins that are user-specific in a way that can be easily checked by vendors. Such coins, if stolen, are of no value to most other users. This greatly reduces the motivation for theft of coins.

In the first proposal, the broker splits the users into "groups," and gives each user coins whose validity depends on the identity of the group. For example, the broker can give user $U$ coins that satisfy the additional condition $h'(x_1, x_2, \ldots, x_k) = h'(U)$, where hash function $h'$ produces short (e.g. 16-bit) output values that indicate $U$'s group. A vendor can easily check this condition, and reject a coin that is not tendered by a member of the correct group.

The problem with this approach is that if the groups are too large, then a thief can easily find users of the appropriate group who might be willing to buy stolen coins. On the other hand, if the groups are too small (e.g. by placing each user is in his own group), the broker may be forced to precompute a large excess of coins, just to ensure that he has a large enough supply to satisfy each user's unpredictable needs.

In the second proposal, we generalize the notion of a "collision" to more complicated combinatorial structures. Formally, a coin $(x_1, \ldots, x_k)$ will be valid for a user $U$ if the images $y_1 = h(x_1)$, $y_2 = h(x_2)$, $\ldots$, $y_k = h(x_k)$ satisfy the condition

$$y_{i+1} - y_i = d_i \pmod{2^u}$$

for $i = 1, 2, \ldots, k-1$, where

$$(d_1, d_2, \ldots, d_{k-1}) = h'(U)$$

for a suitable auxiliary hash function $h'$. (The original proposal for representing coins as collisions can be viewed as the special case where all the distances $d_i$'s between the $k$ bins are zero.)

To mint coins of this form, the broker fills up most of his bins by randomly tossing balls into them, except that now it is not necessary to have more than one ball per bin. We emphasize that this pre-computation is not user-specific, and the broker does not need to have any prior knowledge of the number of coins that will be requested by each user, since each good ball can be used in a coin for *any* user. After this lengthy pre-computation, the broker can quickly create a coin for any user $U$ by

- Computing $(d_1, \ldots, d_{k-1}) = h'(U)$.
- Picking a random bin index $y_1$. (This bin should have been previously unused as a $y_1$ for another coin, so that $y_1$ can be used as the "identity" of the coin when the broker uses a bit-array to determine which coins have already been redeemed.)
- Computing $y_{i+1} = y_i + d_i \pmod{2^u}$ for $i = 1, 2, \ldots, k-1$,
- Taking a ball $x_1$ out of bin $y_1$, and taking a copy of one ball out of each bin $y_2$, $\ldots$, $y_k$. (If any bin $y_i$ is empty, start over with a new $y_1$.) Note that balls may be re-used in this scheme.
- Producing the ordered $k$-tuple $(x_1, \ldots, x_k)$ as the output coin.

A convenient feature of this scheme is that it is easy to produce a large number of coins for a given user even when the broker's storage device is a magnetic disk with a relatively slow seek time. The idea is based on the observation that if the $y_1$ values for successive coins are consecutive, then so also will be the $y_i$ values for each $i$, $1 < i \leq k$. Therefore, a request for 2500 new coins with $k = 4$ requires only four disk seeks, rather than 10,000 seeks: at 10 milliseconds per seek, this reduces the total seek time from 100 seconds to only 40 milliseconds.

Note that in principle coins produced for different users could re-use the same ball $x_i$. Conceivably, someone could forge a new coin by combining pieces of other coins he has seen. However, he is unlikely to achieve much success by this route unless he sees balls from a significant fraction of all the bins. For example, suppose that there are $2^{31}$ bins, of which the forger has seen a fraction $2^{-10}$ (i.e., he has collected $2^{21}$ balls from coins spent by other users). Then the expected number of coins he can piece together from these balls that satisfy the condition of being a good coin for himself is only $2^{31}(2^{-10})^3 = 2$. (Even if he had 1000 customers for these coins, he would expect to make only 2000 coins total, or two coins per customer on the average.) Thus, we are not too concerned about this sort of "cut-and-paste" forgery.

**Vendor-specific coins**

To further reduce the likelihood that coins will be stolen, the user can give coins to vendors in such a way that each coin can be redeemed only by a small fraction of the vendors. This technique makes a stolen coin less desirable, since it is unlikely to be accepted by a vendor other than the one where it was originally spent. The additional check of validity can be carried out both by the vendor and by the broker. (Having vendor-specific coins is also a major feature of the Millicent [10] scheme.)

The obvious difficulty is that neither the broker nor the user can predict ahead of time which vendors the user will patronize, and it is unreasonable to force the user to purchase in advance coins specific for each possible vendor. Millicent adopts the alternative strategy whereby the user must contact the broker in real-time whenever the user needs coins for a new vendor. (He also needs to contact the broker to return excess unused coins that are specific to that vendor.) We can overcome these problems with an extension of the user-specific scheme described above, in which the user purchases a block of "successive" MicroMint coins.

Intuitively, the idea is the following. Choose a value $v$ (e.g. 1024) less than $u$. Let a $u$-bit bin-index $y$ be divided into a $u - v$-bit upper part $y'$ and a $v$-bit lower part $y''$. We consider that $y'$ specifies a "superbin" index and that $y''$ specifies a bin within that superbin. A user now purchases balls in bulk and makes his own coins. He purchases balls by the superbin, obtaining $2^v$ balls per superbin with one ball in each bin of the superbin. He buy $k$ superbins of balls for $2^v$ cents. A coin from user $U$ is valid for redemption by vendor $V$ if:

$$y'_{i+1} = y'_i + d'_i \pmod{2^{u-v}} \text{ for } i = 1, \ldots, k-1,$$

and

$$y''_{i+1} = y''_i + d''_i \pmod{2^v} \text{ for } i = 1, \ldots, k-1,$$

where

$$h'(U) = (d'_1, \ldots, d'_{k-1})$$

and

$$h''(V) = (d''_1, \ldots, d''_{k-1}) .$$

The broker chooses the next available superbin as the first superbin to give the user; the other superbins are then uniquely determined by the differences $\{d'_i\}$ defined by the user's identity and the choice of the first superbin. Analogously, to make a coin for a particular vendor the user chooses a ball from the next bin from his first superbin, and must use balls from bins in the other superbins that are then uniquely determined by the differences $\{d''_i\}$ defined by the vendor's identity and the choice of the first bin. Note that balls from the first superbin are used only once, to permit detection of double-spending, whereas balls from the other superbins may appear more than once (in coins paid to different vendors), or not at all. It may be difficult for a broker to create superbins that are perfectly full even if he throws more balls. He might sell superbins that are almost full, but then a user may have difficulty producing some coins for some vendors. To compensate, the broker can reduce the price by one cent for each empty bin sold.

### Simultaneously generating balls for multiple months

Our major line of defense against large-scale forgery is the fact that the broker can compute coins in advance, whereas a forgery attempt can only be started once the new validity condition for the current month is announced. We now describe a technique whereby computing the balls for a single month's coins takes eight months, but the broker doesn't fall behind because he can generate balls for eight future months concurrently. The forger will thus have the dual problems of starting late and being too slow, even if he uses the same computational resources as the real broker.

In this method, the broker changes the monthly validity criterion, not by changing the hash function $h$, but by announcing each month a new value $z$ such that ball $x$ is good when the high-order $t$ bits of $h(x)$ are equal to $z$. The broker randomly and secretly chooses in advance the values $z$ that will be used for each of the next eight months. Tossing a ball still means performing one hash function computation, but the tossed ball is potentially "good" for any of the next eight months, and it is trivial for the broker to determine if this is the case. In contrast, the forger only knows the current value of $z$, and can not afford to memorize all the balls he tosses, since memory is relatively expensive and only a tiny fraction (e.g., $2^{-21}$ in our running example) of the balls are considered "good" at any given month.

We now describe a convenient way of carrying out this calculation. Assume that at the beginning of the month $j$, the broker has all of the balls needed for month $j$, 7/8 of the balls needed for month $j+1$, 6/8 of the balls needed for month $j+2$, ..., and 1/8 of the balls needed in for month $j+7$. During month $j$, the broker tosses balls by randomly picking $x$ values, calculating $y = h(x)$, and checking whether the top-most $t$ bits of $y$ are equal to any of the $z$ values to be used in months $j+1, \ldots, j+8$. To slow the rate at which he generates good balls for each upcoming month, he increases $n$ and $t$ each by three. After the month-long computation, we expect him to have all the coins he needs for month $j+1$, 7/8 of the coins he needs for month $j+2$, and so on; this is the desired "steady-state" situation. The broker needs four times as much storage to hold the balls generated for future months, but balls for future months can be temporarily stored on inexpensive magnetic tapes because he doesn't need to respond quickly to user requests for those coins yet.

### Hidden Predicates

The "hidden predicate" technique for defeating forgers works as follows. We choose $m > n$, and require each $m$-bit pre-image to satisfy a number of hidden predicates. The hidden predicates should be such that generating pre-images satisfying the predicates is easy (if you know the predicate). To generate an $x_i$, one can pick its last $n$ bits randomly, and define the $j$-th bit of $x_i$, for $j = m-n, \ldots, 1$, to be the $j$-th hidden predicate applied to bits $j+1, \ldots, m$ of $x_i$. The hidden predicates must be balanced and difficult to learn from random examples. Suggestions of hard-to-learn predicates exist in the learning-theory literature. For example the parity/majority functions of Blum et al.[4] (which are the exclusive-or of some of the input bits together with the majority function on a disjoint set of input bits) are interesting, although slightly more complicated functions may be appropriate in this application when word lengths are short. With $m - n = 32$, the broker can have one hidden predicate for each day of the month. He could reveal a new predicate each day, and ask vendors to check that the coins they receive satisfy these predicates (otherwise the coins will not be accepted by the broker). This would not affect the validity of legitimate coins already in circulation, but makes forgery extremely difficult, since the would-be forger would have to discard much of his precomputation work as each new predicate is revealed. We feel that such techniques are strongly advisable in MicroMint.

**Other Extensions**

Peter Wayner (private communication) has suggested a variation on MicroMint in which coins of different values are distinguished by publicly-known predicates on the $x$-values.

## 5 Relationship to Other Micropayment Schemes

In this section we compare our proposals to the Millicent[10], NetBill [1], NetCard [2], and Pederson [14] micropayment schemes.

NetBill offers a number of advanced features (such as electronic purchase orders and encryption of purchased information), but it is relative expensive: digital signatures are heavily used and the NetBill server is involved in each payment.

Millicent uses hash functions extensively, but the broker must be on-line whenever the user wishes to interact with a new vendor. The user buys vendor-specific scrip from the broker. For applications such as web browsing, where new user-vendor relationships are continually being created, Millicent can place a heavy real-time burden on the broker. Compared to Millicent, both PayWord and MicroMint enable the user to generate vendor-specific "scrip" without any interaction with the broker, and without the overhead required in returning unused vendor-specific scrip. Also, PayWord is a credit rather than debit scheme.

Anderson, Manifavas, and Sutherland [2] have developed a micropayment system, "NetCard," which is very similar to PayWord in that it uses chains of hash values with a digitally signed root. (The way hash chains are created differs in a minor way.) However, in their proposal, it is the bank rather than the user who prepares the chain and signs the root, which adds to the overall burden of the bank. This approach prevents the user from creating new chains, although a NetCard user could spend a single chain many times. Compared to PayWord, NetCard is debit-based, rather than credit-based. We have heard that a patent has been applied for on the NetCard system.

Torben Pedersen outlines a micropayment proposal[14] that is also based on hash chains. His motivating application was for incremental payment of telephone charges. His paper does not provide much detail on many points (e.g. whether the system is credit or debit-based, how to handle exceptions, whether chains are vendor-specific, and other auxiliary security-related matters). The CAFE project has filed for a patent on what we believe is an elaboration of Pedersen's idea. (The details off the CAFE scheme are not available to us.)

Similarly following Pedersen's exposition, the iKP developers Hauser, Steiner, and Waidner have independently adopted a similar approach [8].

## 6 Conclusions and Discussion

We have presented two new micropayment schemes which are exceptionally economical in terms of the number of public-key operations employed. Furthermore, both schemes are *off-line* from the broker's point of view.

## References

[1] The NetBill Electronic Commerce Project, 1995. `http://www.ini.cmu/NETBILL/home.html`.

[2] Ross Anderson, Harry Manifavas, and Chris Sutherland. A practical electronic cash system, 1995. Available from author: `Ross.Anderson@cl.cam.ac.uk`.

[3] Matt Blaze, Whitfield Diffie, Ronald L. Rivest, Bruce Schneier, Tsutomu Shimomura, Eric Thompson, and Michael Wiener. Minimal key lengths for symmetric ciphers to provide adequate commercial security: A report by an ad hoc group of cryptographers and computer scientists, January 1996. Available at `http://www.bsa.org`.

[4] Avrim Blum, Merrick Furst, Michael Kearns, and Richard J. Lipton. Cryptographic primitives based on hard learning problems. In Douglas R. Stinson, editor, *Proc. CRYPTO 93*, pages 278–291. Springer, 1994. Lecture Notes in Computer Science No. 773.

[5] Shimon Even, Oded Goldreich, and Silvio Micali. On-line/off-line digital signatures. In G. Brassard, editor, *Proc. CRYPTO 89*, pages 263–277. Springer-Verlag, 1990. Lecture Notes in Computer Science No. 435.

[6] Phillip Hallam-Baker. W3C payments resources, 1995. `http://www.w3.org/hypertext/WWW/Payments/overview.html`.

[7] Neil M. Haller. The S/KEY one-time password system. In *ISOC*, 1994.

[8] Ralf Hauser, Michael Steiner, and Michael Waidner. Micro-Payments based on iKP, December 17, 1995. Available from authors. `sti@zurich.ibm.com`.

[9] Leslie Lamport. Password authentication with insecure communication. *Communications of the ACM*, 24(11):770–771, November 1981.

[10] Mark S. Manasse. Millicent (electronic microcommerce), 1995. `http://www.research.digital.com/SRC/personal/Mark_Manasse/uncommon/ucom.html`.

[11] Ralph C. Merkle. A certified digital signature. In G. Brassard, editor, *Proc. CRYPTO 89*, pages 218–238. Springer-Verlag, 1990. Lecture Notes in Computer Science No. 435.

[12] Silvio Micali. Efficient certificate revocation. Technical Report TM-542b, MIT Laboratory for Computer Science, March 22, 1996.

[13] National Institute of Standards and Technology (NIST). *FIPS Publication 180: Secure Hash Standard (SHS)*, May 11, 1993.

[14] Torben P. Pedersen. Electronic payments of small amounts. Technical Report DAIMI PB-495, Aarhus University, Computer Science Department, Århus, Denmark, August 1995.

[15] Ronald L. Rivest. The MD5 message-digest algorithm. Internet Request for Comments, April 1992. RFC 1321.

[16] Bruce Schneier. *Applied Cryptography (Second Edition)*. John Wiley & Sons, 1996.

[17] Adi Shamir. Fast signature screening. CRYPTO '95 rump session talk; to appear in RSA Laboratories' *CryptoBytes*.

[18] Peter Wayner. *Digital Cash: Commerce on the Net*. Academic Press, 1996.

# Techniques for data hiding

by W. Bender
D. Gruhl
N. Morimoto
A. Lu

***Data hiding, a form of steganography, embeds data into digital media for the purpose of identification, annotation, and copyright. Several constraints affect this process: the quantity of data to be hidden, the need for invariance of these data under conditions where a "host" signal is subject to distortions, e.g., lossy compression, and the degree to which the data must be immune to interception, modification, or removal by a third party. We explore both traditional and novel techniques for addressing the data-hiding process and evaluate these techniques in light of three applications: copyright protection, tamper-proofing, and augmentation data embedding.***

Digital representation of media facilitates access and potentially improves the portability, efficiency, and accuracy of the information presented. Undesirable effects of facile data access include an increased opportunity for violation of copyright and tampering with or modification of content. The motivation for this work includes the provision of protection of intellectual property rights, an indication of content manipulation, and a means of annotation. Data hiding represents a class of processes used to embed data, such as copyright information, into various forms of media such as image, audio, or text with a minimum amount of perceivable degradation to the "host" signal; i.e., the embedded data should be invisible and inaudible to a human observer. Note that data hiding, while similar to compression, is distinct from encryption. Its goal is not to restrict or regulate access to the host signal, but rather to ensure that embedded data remain inviolate and recoverable.

Two important uses of data hiding in digital media are to provide proof of the copyright, and assurance of content integrity. Therefore, the data should stay hidden in a host signal, even if that signal is subjected to manipulation as degrading as filtering, resampling, cropping, or lossy data compression. Other applications of data hiding, such as the inclusion of augmentation data, need not be invariant to detection or removal, since these data are there for the benefit of both the author and the content consumer. Thus, the techniques used for data hiding vary depending on the quantity of data being hidden and the required invariance of those data to manipulation. Since no one method is capable of achieving all these goals, a class of processes is needed to span the range of possible applications.

The technical challenges of data hiding are formidable. Any "holes" to fill with data in a host signal, either statistical or perceptual, are likely targets for removal by lossy signal compression. The key to successful data hiding is the finding of holes that are not suitable for exploitation by compression algorithms. A further challenge is to fill these holes with data in a way that remains invariant to a large class of host signal transformations.

0018-8670/96/$5.00 © 1996 IBM

## Features and applications

Data-hiding techniques should be capable of embedding data in a host signal with the following restrictions and features:

[redacted]

1. The host signal should be nonobjectionally degraded and the embedded data should be minimally perceptible. (The goal is for the data to remain *hidden*. As any magician will tell you, it is possible for something to be hidden while it remains in plain sight; you merely keep the person from looking at it. We will use the words *hidden*, *inaudible*, *imperceivable*, and *invisible* to mean that an observer does not notice the presence of the data, even if they are perceptible.)
2. The embedded data should be directly encoded into the media, rather than into a header or wrapper, so that the data remain intact across varying data file formats.
3. The embedded data should be immune to modifications ranging from intentional and intelligent attempts at removal to anticipated manipulations, e.g., channel noise, filtering, resampling, cropping, encoding, lossy compressing, printing and scanning, digital-to-analog (D/A) conversion, and analog-to-digital (A/D) conversion, etc.
4. Asymmetrical coding of the embedded data is desirable, since the purpose of data hiding is to keep the data in the host signal, but not necessarily to make the data difficult to access.
5. Error correction coding[1] should be used to ensure data integrity. It is inevitable that there will be some degradation to the embedded data when the host signal is modified.
6. The embedded data should be self-clocking or arbitrarily re-entrant. This ensures that the embedded data can be recovered when only fragments of the host signal are available, e.g., if a sound bite is extracted from an interview, data embedded in the audio segment can be recovered. This feature also facilitates automatic decoding of the hidden data, since there is no need to refer to the original host signal.

**Applications.** Trade-offs exist between the quantity of embedded data and the degree of immunity to host signal modification. By constraining the degree of host signal degradation, a data-hiding method can operate with either high embedded data rate, or high resistance to modification, but not both. As one increases, the other must decrease. While this can be shown mathematically for some data-hiding systems such as a spread spectrum, it seems to hold true for all data-hiding systems. In any system, you can trade bandwidth for robustness by exploiting redundancy. The quantity of embedded data and the degree of host signal modification vary from application to application. Consequently, different techniques are employed for different applications. Several prospective applications of data hiding are discussed in this section.

[redacted]

An application that requires a minimal amount of embedded data is the placement of a digital watermark. The embedded data are used to place an indication of ownership in the host signal, serving the same purpose as an author's signature or a company logo.

**Trade-offs exist between the quantity of data and the immunity to modification.**

Since the information is of a critical nature and the signal may face intelligent and intentional attempts to destroy or remove it, the coding techniques used must be immune to a wide variety of possible modifications.

A second application for data hiding is tamper-proofing. It is used to indicate that the host signal has been modified from its authored state. Modification to the embedded data indicates that the host signal has been changed in some way.

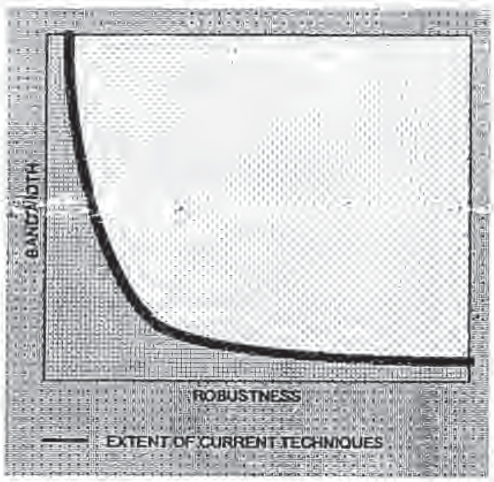
A third application, feature location, requires more data to be embedded. In this application, the embedded data are hidden in specific locations within an image. It enables one to identify individual content features, e.g., the name of the person on the left versus the right side of an image. Typically, feature location data are not subject to intentional removal. However, it is expected that the host signal might be subjected to a certain degree of modification, e.g., images are routinely modified by scaling, cropping, and tone-scale enhancement. As a result, feature location data-hiding techniques must be immune to geometrical and nongeometrical modifications of a host signal.

Image and audio captions (or annotations) may require a large amount of data. Annotations often travel separately from the host signal, thus requiring additional channels and storage. Annotations stored in file headers or resource sections are often lost if the file format is changed, e.g., the annotations created in a Tagged Image File Format (TIFF) may not be present when the image is transformed to a Graphic Interchange Format (GIF). These problems are resolved by embedding annotations directly into the data structure of a host signal.

**Prior work.** Adelson[2] describes a method of data hiding that exploits the human visual system's varying sensitivity to contrast versus spatial frequency. Adelson substitutes high-spatial frequency image data for *hidden* data in a pyramid-encoded still image. While he is able to encode a large amount of data efficiently, there is no provision to make the data immune to detection or removal by typical manipulations such as filtering and rescaling. Stego,[3] one of several widely available software packages, simply encodes data in the least-significant bit of the host signal. This technique suffers from all of the same problems as Adelson's method but creates an additional problem of degrading image or audio quality. Bender[4] modifies Adelson's technique by using *chaos* as a means to encrypt the embedded data, deterring detection, but providing no improvement to immunity to host signal manipulation. Lippman[5] hides data in the chrominance channel of the National Television Standards Committee (NTSC) television signal by exploiting the temporal over-sampling of color in such signals. Typical of Enhanced Definition Television Systems, this method encodes a large amount of data, but the data are lost to most recording, compression, and transcoding processes. Other techniques, such as Hecht's Data-Glyph,[6] which adds a *bar code* to images, are engineered in light of a predetermined set of geometric modifications.[7] Spread-spectrum,[8-11] a promising technology for data hiding, is difficult to intercept and remove but often introduces perceivable distortion into the host signal.

**Problem space.** Each application of data hiding requires a different level of resistance to modification and a different embedded data rate. These form the conceptual data-hiding problem space (see Figure 1). There is an inherent trade-off between bandwidth and "robustness," or the degree to which the data are immune to attack or transformations that occur to the host signal through normal usage, e.g., compression, resampling, etc. The more data to be hidden, e.g., a caption for a photograph, the less secure the encoding. The less data to be hidden, e.g., a watermark, the more secure the encoding.

**Figure 1 Conceptual data-hiding problem space**



## Data hiding in still images

Data hiding in still images presents a variety of challenges that arise due to the way the human visual system (HVS) works and the typical modifications that images undergo. Additionally, still images provide a relatively small host signal in which to hide data. A fairly typical 8-bit picture of 200 × 200 pixels provides approximately 40 kilobytes (kB) of data space in which to work. This is equivalent to only around 5 seconds of telephone-quality audio or less than a single frame of NTSC television. Also, it is reasonable to expect that still images will be subject to operations ranging from simple affine transforms to nonlinear transforms such as cropping, blurring, filtering, and lossy compression. Practical data-hiding techniques need to be resistant to as many of these transformations as possible.

Despite these challenges, still images are likely candidates for data hiding. There are many attributes of the HVS that are potential candidates for exploitation in a data-hiding system, including our varying sensitivity to contrast as a function of spatial frequency and the masking effect of edges (both in luminance and

**Figure 2** A single iteration in the Patchwork method (photograph courtesy of Webb Chapel)

chrominance). The HVS has low sensitivity to small changes in luminance, being able to perceive changes of no less than one part in 30 for random patterns. However, in uniform regions of an image, the HVS is more sensitive to the change of the luminance, approximately one part in 240. A typical CRT (cathode ray tube) display or printer has a limited dynamic range. In an image representation of one part in 256, e.g., 8-bit gray levels, there is potentially room to hide data as pseudorandom changes to picture brightness. Another HVS "hole" is our relative insensitivity to very low spatial frequencies such as continuous changes in brightness across an image, i.e., vignetting. An additional advantage of working with still images is that they are noncausal. Data-hiding techniques can have access to any pixel or block of pixels at random.

Using these observations, we have developed a variety of techniques for placing data in still images. Some techniques are more suited to dealing with small amounts of data, while others to large amounts. Some techniques are highly resistant to geometric modifications, while others are more resistant to nongeometric modifications, e.g., filtering. We present methods that explore both of these areas, as well as their combination.

## Low bit-rate data hiding

With low bit-rate encoding, we expect a high level of robustness in return for low bandwidth. The emphasis is on resistance to attempts of data removal by a third party. Both a statistical and a perceptual technique are discussed in the next sections on Patchwork, texture, and applications.

## Patchwork: A statistical approach

The statistical approach, which we refer to as *Patchwork*, is based on a pseudorandom, statistical process. Patchwork invisibly embeds in a host image a specific statistic, one that has a Gaussian distribution. Figure 2 shows a single iteration in the Patchwork method. Two patches are chosen pseudorandomly, the first A, the second B. The image data in patch A are lightened while the data in patch B are darkened (exaggerated for purposes of this illustration). This unique statistic indicates the presence or absence of a signature. Patchwork is independent of the contents of the host image. It shows reasonably high resistance to most nongeometric image modifications.

For the following analysis, we make the following simplifying assumptions (these assumptions are not limiting, as is shown later): We are operating in a 256 level, linearly quantized system starting at 0; all brightness levels are equally likely; all samples are independent of all other samples.

The Patchwork algorithm proceeds as follows: take any two points, $A$ and $B$, chosen at random in an image. Let $a$ equal the brightness at point $A$ and $b$ the brightness at point $B$. Now, let

$$S = a - b \tag{1}$$

The *expected* value of $S$ is 0, i.e., the average value of $S$ after repeating this procedure a large number of times is *expected* to be 0.

Although the *expected* value is 0, this does not tell us much about what $S$ will be for a specific case. This is because the variance is quite high for this procedure. The variance of $S$, $\sigma$, is a measure of how tightly samples of $S$ will cluster around the expected value of 0. To compute this, we make the following observation: Since $S = a - b$ and $a$ and $b$ are assumed independent, $\sigma_S^2$ can be computed as follows (this, and all other probability equations are from Drake[12]):

$$\sigma_S^2 = \sigma_a^2 + \sigma_b^2 \qquad (2)$$

where $\sigma_a^2$ for a uniform $S$ is:

$$\sigma_a^2 = 5418 \qquad (3)$$

Now, $\sigma_a^2 = \sigma_b^2$ since $a$ and $b$ are samples from the same set, taken with replacement. Thus:

$$\sigma_S^2 = 2 \times \sigma_a^2 = 2 \times \frac{(255-0)^2}{12} \approx 10836 \qquad (4)$$

which yields a standard deviation $\sigma_S \approx 104$. This means that more than half the time, $S$ will be greater than 43 or less than $-43$. Assuming a Gaussian clustering, a single iteration does not tell us much. However, this is not the case if we perform the procedure many times.

Let us repeat this procedure $n$ times, letting $a_i$ and $b_i$ be the values $a$ and $b$ take on during the $i$th iteration, $S_i$. Now let $S_n$ be defined as:

$$S_n = \sum_{i=1}^{n} S_i = \sum_{i=1}^{n} a_i - b_i \qquad (5)$$

The *expected* value of $S_n$ is:

$$S_n = n \times S = n \times 0 = 0 \qquad (6)$$

This makes intuitive sense, since the number of times $a_i$ is greater than $b_i$ should be offset by the number of times the reverse is true. Now the variance is:

$$\sigma_{S_n}^2 = n \times \sigma_S^2 \qquad (7)$$

And the standard deviation is:

$$\sigma_{S_n} = \sqrt{n} \times \sigma = \sqrt{n} \times 104 \qquad (8)$$

**Table 1** **Degree of certainty of encoding given deviation from that expected in a Gaussian distribution ($\delta = 2$)**

| Standard Deviations Away | Certainty | n |
|---|---|---|
| 0 | 50.00% | 0 |
| 1 | 84.13% | 679 |
| 2 | 97.8[illegible]% | 2713 |
| 3 | 99.87% | 6104 |

Now, we can compute $S_{10000}$ for a picture, and if it varies by more than a few standard deviations, we can be fairly certain that this did not happen by chance. In fact, since as we will show later $S'_n$ for large $n$ has a Gaussian distribution, a deviation of even a few $\sigma_S$s indicates to a high degree of certainty the presence of encoding (see Table 1).

The Patchwork method artificially modifies $S$ for a given picture, such that $S'_n$ is many deviations away from expected. To encode a picture, we:

1. Use a specific key for a known pseudorandom number generator to choose $(a_i, b_i)$. This is important, because the encoder needs to visit the same points during decoding.
2. Raise the brightness in the patch $a_i$ by an amount $\delta$, typically in the range of 1 to 5 parts in 256.
3. Lower the brightness in $b_i$ by this same amount $\delta$ (the amounts do not have to be the same, as long as they are in opposite directions).
4. Repeat this for $n$ steps ($n$ typically ~10 000).

Now, when decoded, $S'_n$ will be:

$$S'_n = \sum_{i=1}^{n} (a_i + \delta) - (b_i - \delta) \qquad (9)$$

or:

$$S'_n = 2\delta n + \sum_{i=1}^{n} (a_i - b_i) \qquad (10)$$

So each step of the way we accumulate an expectation of $2 \times \delta$. Thus after $n$ repetitions, we expect $S'_n$ to be:

**Figure 3** As δ or $n$ increases, the distribution of $S'_n$ shifts further to the right.

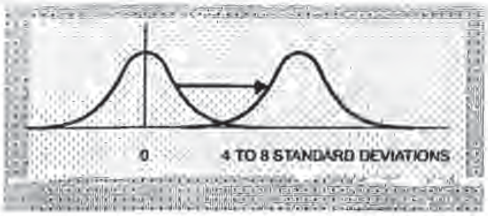


**Figure 4** The contour of a patch largely determines which frequencies will be modified by the application of Patchwork.

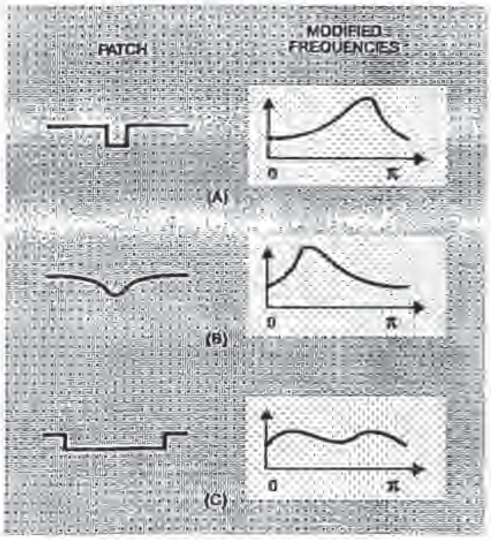
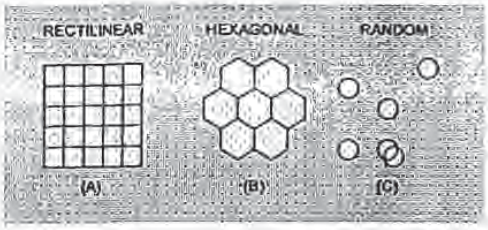


**Figure 5** Patch placement affects patch visibility.



$$\frac{2\delta n}{\sigma_{S'_n}} \approx 0.028\delta\sqrt{n} \qquad (11)$$

As $n$ or δ increases, the distribution of $S'_n$ shifts over to the right (Figure 3 and Table 1). In Figure 3, as δ or $n$ increases, the distribution of $S'_n$ shifts further to the right. If we shift it far enough, any point that is likely to fall into one distribution is highly unlikely to be near the center of the other distribution.

While this basic method works well by itself, we have made a number of modifications to improve performance including:

1. Treating *patches* of several points rather than single points. This has the effect of shifting the noise introduced by Patchwork into the lower spatial frequencies, where it is less likely to be removed by lossy compression and typical Finite Impulse Response (FIR) filters.
2. Making Patchwork more robust by using a combination with either affine coding (described later) or some heuristic based upon feature recognition (e.g., alignment using the interocular line of a face). Patchwork decoding is sensitive to affine transformations of the host image. If the points in the picture visited during encoding are offset by translation, rotation, or scaling before decoding, the code is lost.
3. Taking advantage of the fact that Patchwork is fairly resistant to cropping. By disregarding points outside of the known picture area, Patchwork degrades in accuracy approximately as the log of the picture size. Patchwork is also resistant to gamma and tone scale correction since values of comparable luminance move roughly the same way under such modifications.

**Patch shape.** The shape of the patches deserves some comment. Figure 4 shows three possible one-dimensional patch shapes, and next to them a very approximate spectrum of what a line with these patches dropped onto it pseudorandomly would look like. In Figure 4A, the patch is very small, with sharp edges. This results in the majority of the energy of the patch being concentrated in the high frequency portion of the image spectrum. This makes the distortion hard to see, but also makes it a likely candidate for removal by lossy compressors. If one goes to the other extreme, as in Figure 4B, the majority of the information is contained in the low-frequency spectrum. The last choice, Figure 4C shows a wide, sharp-edged patch, which tends to distribute the energy around the entire frequency spectrum.

The optimal choice of patch shape is dependent upon the expected image modifications. If JPEG (Joint Photographic Experts Group) encoding is likely, then a patch that places its energy in the low frequencies is preferable. If contrast enhancement is to be done, placing energy in higher frequencies would be better. If the potential image modifications are unknown, then spreading the patch energy across the spectrum would make sense.

The arrangement of patches has an impact on patch visibility. For illustration, three possibilities are considered (Figure 5). The simplest method is shown in Figure 5A, a simple rectilinear lattice. While simple, this arrangement is often a poor choice if a high $n$ is to be used. As the grid is filled in, continuous edges of gradient are formed. The HVS is very sensitive to such edges. A second choice, Figure 5B breaks this symmetry by using hexagons for the patch shape. A preferred solution, shown in Figure 5C, is a completely random placement of patches. An intelligent selection of patch shape in both the horizontal and vertical dimensions will enhance the effectiveness of patchwork for a given picture.

**Uniformity.** A simplifying assumption of a uniform luminance histogram was made above, but this is not a requirement of Patchwork. The only assumption Patchwork makes is that the *expected* value of $S_i = a_i - b_i$ is zero.

It can be shown that this condition is always met through the following argument:

1. Let $a_r$ be the time reversed series of $a$.
2. $A_r = A^*$ by definition ($A^*$ is the complex conjugate of $A$).
3. $F(a * a_r) = AA^*$ ($F$ is the Fourier transform).
4. $AA^*$ is everywhere real by definition of the complex conjugate.
5. $F^{-1}(AA^*)$ is even by definition.
6. Even sequences are symmetric around zero by definition.

An image histogram (Figure 6, top) is a somewhat random distribution. The result of taking the complex conjugate (Figure 6, bottom) is symmetric around zero.

**Figure 6** A histogram of Figure 2 and its autocorrelation

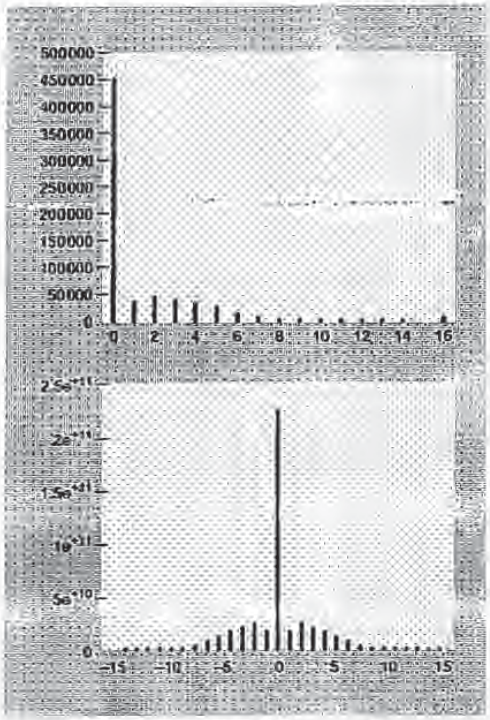
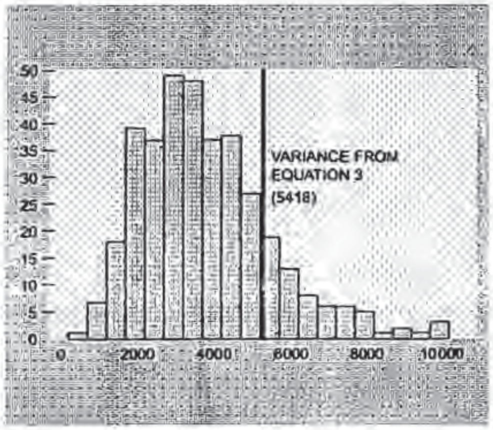


**Figure 7** A histogram of the variance of the luminance of 365 Associated Press photos from March 1996



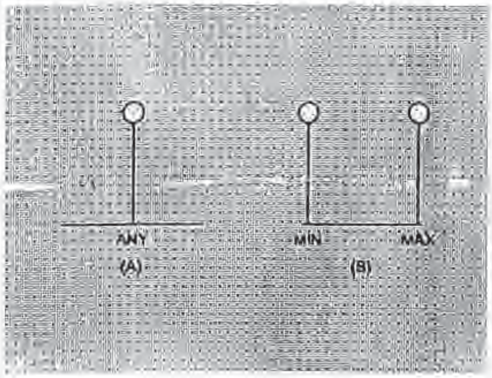


**Figure 8** Histograms of pictures with minimum (A) and maximum (B) variance

**Variance.** When searching through a large number of images with data embedded using the Patchwork method, such as when a robot is looking for copyright violations on the Internet World Wide Web (WWW), the use of a generic estimation of variance is desirable. This avoids the necessity of calculating the variance of every image. Suspect images can then be examined thoroughly.

When, in the analysis above, the number of points needed in Equation 3 was computed, the variance of the luminance was assumed to be 5418. This assumption turns out to be higher than the average observed values (see Figure 7). The question is, then, what value should be used.

An examination of the variance of 365 Associated Press photos from March 1996 yielded an average value of 3877.4 and a distribution that can be seen in Figure 7. While some pictures do have variances as high as two-thirds of the maximum, most are clustered around the lower variance values. Thus, 5418, the estimate derived from the uniformity assumption, is a conservative but reasonable value to use for a generic picture.

A minimum value is that for a solid color picture (Figure 8A). This has a variance of 0, a standard deviation of 0, and thus works very well for Patchwork, since any modification is evident. The other extreme is that of a two-color, black and white picture. For these, the variance is:

$$\frac{(0-127)^2}{2}+\frac{(255-127.5)^2}{2}=16256 \qquad (12)$$

These two values, 0 and 16256, define the extremes of the variance to consider when calculating the likelihood that a picture is encoded. What is the correct assumption to use for a given picture? The actual variance of the picture being examined is a sensible choice, since in most cases Patchwork will increase the variance only slightly. (This depends on the size and depth of the patch, the number of patches, and the histogram of the original image.) However, if a large number of pictures are to examined, a generic value is a practical choice.

**Summary.** There are several limitations inherent to the Patchwork technique. The first is the extremely low embedded data rate it yields, usually a one-bit signature per image. This limits its usefulness to low bit-rate applications such as the digital watermark. Second, it is necessary to *register* where the pixels in the image lie. While a number of methods have been investigated, it is still somewhat difficult to decode the image in the presence of severe affine transformations. These disadvantages aside, without the key for the pseudorandom number generator, it is extremely difficult to remove the Patchwork coding without degrading the picture beyond recognition.

The Patchwork method is subject to cryptographic attack if it is used to encode a large number of identically sized images using the same key. If the images are averaged together, the *patches* will show up as lighter or darker than average regions. This weakness is a common one in cryptography, and points to the truism that for a static key, as the amount of traffic increases, it becomes easier to "crack" the encryption. One solution is to use multiple pseudorandom patterns for the patches. Even the use of just two keys, while increasing decoding time, will make Patchwork much more robust to attack. Another solution is to use the same pattern, but to reverse the polarity of the patches. Both solutions deter cryptographic attack by averaging.

## Texture Block Coding: A visual approach

A second method for low bit-rate data hiding in images is *Texture Block Coding*. This method hides data within the continuous random texture patterns of a picture. The Texture Block Coding technique is

**Figure 9** Texture Block Coding example (photograph courtesy of Webb Chapel)

implemented by copying a region from a random texture pattern found in a picture to an area that has similar texture. This results in a pair of identically textured regions in the image (see Figure 9).

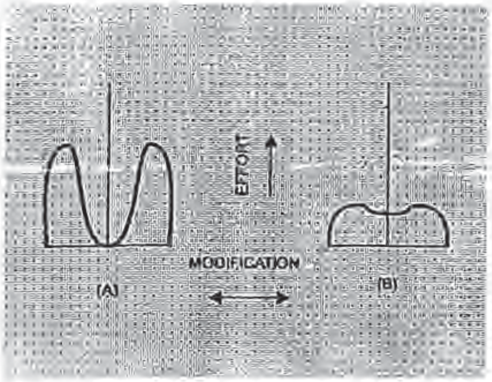
These regions can be detected as follows:

1. Autocorrelate the image with itself. This will produce peaks at every point in the autocorrelation where identical regions of the image overlap. If large enough areas of an image are copied, this will produce an additional large autocorrelation peak at the correct alignment for decoding.
2. Shift the image as indicated by the peaks in Step 1. Now subtract the image from its shifted copy, padding the edges with zeros as needed.
3. Square the result and threshold it to recover only those values quite close to zero. The copied region will be visible as these values.

Since the two regions are identical, they are modified in the same way if the picture is uniformly transformed. By making the regions reasonably large, the inner part of the block changes identically under most nongeometric transformations. In our experiments, coded 16 × 16 pixel blocks can be decoded when the picture is subjected to a combination of filtering, compression, and rotation.

Texture Block Coding is not without its disadvantages. Currently it requires a human operator to choose the source and destination regions, and to evaluate the visual impact of the modifications on the image. It should be possible to automate this process by allowing a computer to identify possible texture regions in the image to copy from and paste to. However, this technique will not work on images that lack moderately large areas of continuous texture from which to draw.

**Figure 10** Characterizing the difference between tamper-proofing and other data-hiding techniques



Future research in this area includes the possibility of cutting and pasting blocks from only part of the image frequency spectrum (this would allow less noticeable blocks to be moved around, and a final encoding that is considerably more robust to various image compression algorithms) along with automatic texture region selection and analysis of perceivability of the final result.

## High bit-rate coding

High bit-rate methods can be designed to have minimal impact upon the perception of the host signal, but they do not tend to be immune to image modifications. In return, there is an expectation that a relatively large amount of data are able to be encoded. The most common form of high bit-rate encoding is the replacement of the least significant luminance bit of image data with the embedded data. Other techniques include the introduction of high-frequency, low-amplitude noise and the use of direct sequence spread spectrum coding. All high bit-rate methods can be made more robust through the use of error-correction coding, at the expense of data rate. High bit-rate codes are only appropriate where it is reasonable to expect that a great deal of control will be maintained over the images.

Individually, none of the known techniques for data hiding are resistant to all possible transforms or combinations of transforms. In combination, often one technique can supplement another. Supplementary techniques are particularly important for recovery from geometric modifications such as affine transformations, and maintaining synchronization for spread-spectrum encoding.

**Affine coding.** Some of the data-hiding techniques, such as Patchwork, are vulnerable to affine transforms. It makes sense to develop methods that can be used to facilitate the recovery of embedded data after affine application. *Affine coding* is one such method: A predefined reference pattern is embedded into a host image using any of the high bit-rate coding techniques. Estimation of geometric transformation of the image is achieved by comparing the original shape, size, and orientation of the reference pattern to that found in the transformed image. Since affine transforms are linear, the inverse transform can be applied to recover the original image. Once this is done, the image is ready for further extraction of embedded data.

## Applications

Placing data in images is useful in a variety of applications. We highlight below four applications that differ in the quantity of data to be embedded and the type of transforms to which the data are likely to be subjected.

**Digital watermark.** The objective of a digital watermark is to place an indelible mark on an image. Usually, this means encoding only a handful of bits, sometimes as few as one. This "signature" could be used as a means of tracing the distribution of images for an on-line news service and for photographers who are selling their work for digital publication. One could build a digital camera that places a watermark on every photograph it takes. Theoretically, this would allow photographers to employ a "web-searching agent" to locate sites where their photographs appear.

It can be expected that if information about legal ownership is to be included in an image, it is likely that someone might want to remove it. A requirement of a digital watermark is that it must be difficult to remove. Both the Patchwork and Texture Block Coding techniques show promise as digital watermarks. Patchwork, being the more secure of the two, answers the question "Is this my picture?" Texture Block Coding, which can be made readily accessible to the public, answers the question "Whose picture is this?"

**Tamper-proofing.** The objective of tamper-proofing is to answer the question, "Has this image been modified?" Tamper-proofing techniques are related, but distinct from the other data-hiding technologies. What differentiates them is the degree to which information is secured from the host signal. In Figure 10, the difference between tamper-proofing and other data-hiding techniques is characterized. Figure 10A illustrates that data hiding requires a deep *information well* that is resilient to large displacements. Figure 10B illustrates that tamper-proofing requires a shallow well that is only resilient to small displacements, but is triggered by large displacements. Most data-hiding techniques attempt to secure data in the face of all modifications. Tamper-proofing techniques must be resilient to small modifications (e.g., cropping, tone-scale or gamma correction for images or balance or equalization for sounds) but not to large modifications (e.g., removing or inserting people from an image or taking words out of context in an audio recording).

There are several ways to implement tamper-proofing. The easiest way is to encode a check-sum of the image within the image. However, this method is triggered by small changes in the image. This suggests an approach involving a pattern overlaid on the image. The key to a successful overlay is to find a pattern resilient to simple modifications such as filtering and gamma correction, yet is not easily removed. The search for such patterns and other methods of detecting tampering remains an active area of research.

**Feature tagging.** Another application of data hiding is tagging the location of features within an image. Using data hiding it is possible for an editor (or machine) to encode descriptive information, such as the location and identification of features of interest, directly into specific regions of an image. This enables retrieval of the descriptive information wherever the image goes. Since the embedded information is spatially located in the image, it is not removed unless the feature of interest is removed. It also translates, scales, and rotates exactly as the feature of interest does.

This application does not have the same requirements for robustness as the digital watermark. It can be assumed that, since feature location is providing a service, it is unlikely someone will maliciously try to remove the encoded information.

**Embedded captions.** Typical news photograph captions contain one kB of data. Thus embedded captions is a relatively high bit-rate application for data hiding. As with feature tagging, caption data are usually not subject to malicious removal.

While captions are useful by themselves, they become even more useful when combined with feature location. It is then possible for portions of the caption to directly reference items in the picture. Captions can become self-editing once this is done. If an item referenced in the caption is cropped out of the picture, then the reference to that item in the caption can be removed automatically.

## Data hiding in audio

Data hiding in audio signals is especially challenging, because the human auditory system (HAS) operates over a wide dynamic range. The HAS perceives over a range of power greater than one billion to one and a range of frequencies greater than one thousand to one. Sensitivity to additive random noise is also acute. The perturbations in a sound file can be detected as low as one part in ten million (80 dB below ambient level). However, there are some "holes" available. While the HAS has a large dynamic range, it has a fairly small differential range. As a result, loud sounds tend to mask out quiet sounds. Additionally, the HAS is unable to perceive absolute phase, only relative phase. Finally, there are some environmental distortions so common as to be ignored by the listener in most cases.

We exploit many of these traits in the methods we discuss next, while being careful to bear in mind the extreme sensitivities of the HAS.

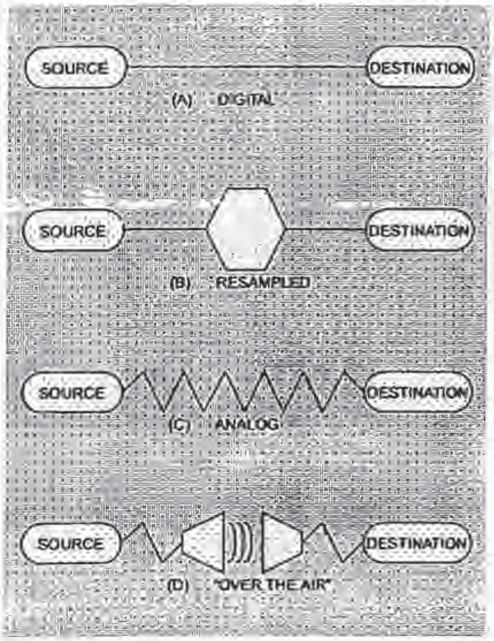
## Audio environments

When developing a data-hiding method for audio, one of the first considerations is the likely environments the sound signal will travel between encoding and decoding. There are two main areas of modification which we will consider. First, the storage environment, or digital representation of the signal that will be used, and second the transmission pathway the signal might travel.

**Digital representation.** There are two critical parameters to most digital audio representations: sample quantization method and temporal sampling rate.

The most popular format for representing samples of high-quality digital audio is a 16-bit linear quantiza-

**Figure 11 Transmission environments**



tion, e.g., Windows Audio-Visual (WAV) and Audio Interchange File Format (AIFF). Another popular format for lower quality audio is the logarithmically scaled 8-bit μ-law. These quantization methods introduce some signal distortion, somewhat more evident in the case of 8-bit μ-law.

Popular temporal sampling rates for audio include 8 kHz (kilohertz), 9.6 kHz, 10 kHz, 12 kHz, 16 kHz, 22.05 kHz, and 44.1 kHz. Sampling rate impacts data hiding in that it puts an upper bound on the usable portion of the frequency spectrum (if a signal is sampled at ~8 kHz, you cannot introduce modifications that have frequency components above ~4 kHz). For most data-hiding techniques we have developed, usable data space increases at least linearly with increased sampling rate.

A last representation to consider is that produced by lossy, perceptual compression algorithms, such as the International Standards Organization Motion Pictures Expert Group—Audio (ISO MPEG-AUDIO) perceptual encoding standard. These representations drastically change the statistics of the signal; they preserve only the characteristics that a listener perceives (i.e., it will sound similar to the original, even if the signal is completely different in a *least squares* sense).

**Transmission environment.** There are many different transmission environments that a signal might experience on its way from encoder to decoder. We consider four general classes for illustrative purposes (see Figure 11). The first is the digital end-to-end environment (Figure 11A). This is the environment of a sound file that is copied from machine to machine, but never modified in any way. As a result, the sampling is exactly the same at the encoder and decoder. This class puts the least constraints on data-hiding methods.

The next consideration is when a signal is resampled to a higher or lower sampling rate, but remains digital throughout (Figure 11B). This transform preserves the absolute magnitude and phase of most of the signal, but changes the temporal characteristics of the signal.

The third case is when a signal is "played" into an analog state, transmitted on a reasonably clean analog line and resampled (Figure 11C). Absolute signal magnitude, sample quantization, and temporal sampling rate are not preserved. In general, phase will be preserved.

The last case is when the signal is "played into the air" and "resampled with a microphone" (Figure 11D). The signal will be subjected to possibly unknown nonlinear modifications resulting in phase changes, amplitude changes, drift of different frequency components, echoes, etc.

Signal representation and transmission pathway must be considered when choosing a data-hiding method. Data rate is very dependent on the sampling rate and the type of sound being encoded. A typical value is 16 bps, but the number can range from 2 bps to 128 bps.

## Low-bit coding

Low-bit coding is the simplest way to embed data into other data structures. By replacing the least significant bit of each sampling point by a coded binary string, we can encode a large amount of data in an audio signal. Ideally, the channel capacity is 1 kb per second (kbps) per 1 kilohertz (kHz), e.g., in a noiseless channel, the bit rate will be 8 kbps in an 8 kHz sampled sequence and 44 kbps in a 44 kHz sampled sequence. In return for this large channel capacity, audible noise is introduced. The impact of this noise is a direct function of the content of the host signal, e.g., crowd noise during a live sports event would mask low-bit encoding noise that would be audible in a string quartet performance. Adaptive data attenuation has been used to compensate this variation.

The major disadvantage of this method is its poor immunity to manipulation. Encoded information can be destroyed by channel noise, resampling, etc., unless it is encoded using redundancy techniques. In order to be robust, these techniques reduce the data rate, often by one to two orders of magnitude. In practice, this method is useful only in closed, digital-to-digital environments.

## Phase coding

The phase coding method works by substituting the phase of an initial audio segment with a reference phase that represents the data. The phase of subsequent segments is adjusted in order to preserve the relative phase between segments.

Phase coding, when it can be used, is one of the most effective coding methods in terms of the signal-to-perceived noise ratio. When the phase relation between each frequency component is dramatically changed, a noticeable phase dispersion will occur. However, as long as the modification of the phase is sufficiently small (sufficiently small depends on the observer; professionals in broadcast radio can detect modifications that are imperceivable to an average observer), an *inaudible* coding can be achieved.

**Procedure.** The procedure for phase coding is as follows:

1. Break the sound sequence $s[i]$, $(0 \le i \le I-1)$, into a series of $N$ short segments, $s_n[i]$ where $(0 \le n \le N-1)$ (Figure 12A, 12B).
2. Apply a K-points discrete Fourier transform (DFT)[13] to $n$-th segment, $s_n[i]$, where $(K = I/N)$, and create a matrix of the phase, $\phi_n(\omega_k)$, and magnitude, $A_n(\omega_k)$ for $(0 \le k \le K-1)$ (Figure 12C).
3. Store the phase difference between each adjacent segment for $(0 \le n \le N-1)$ (Figure 12D):

$$\Delta\phi_{n+1}(\omega_k) = \phi_{n+1}(\omega_k) - \phi_n(\omega_k) \tag{13}$$

Figure 12 Phase coding schematic

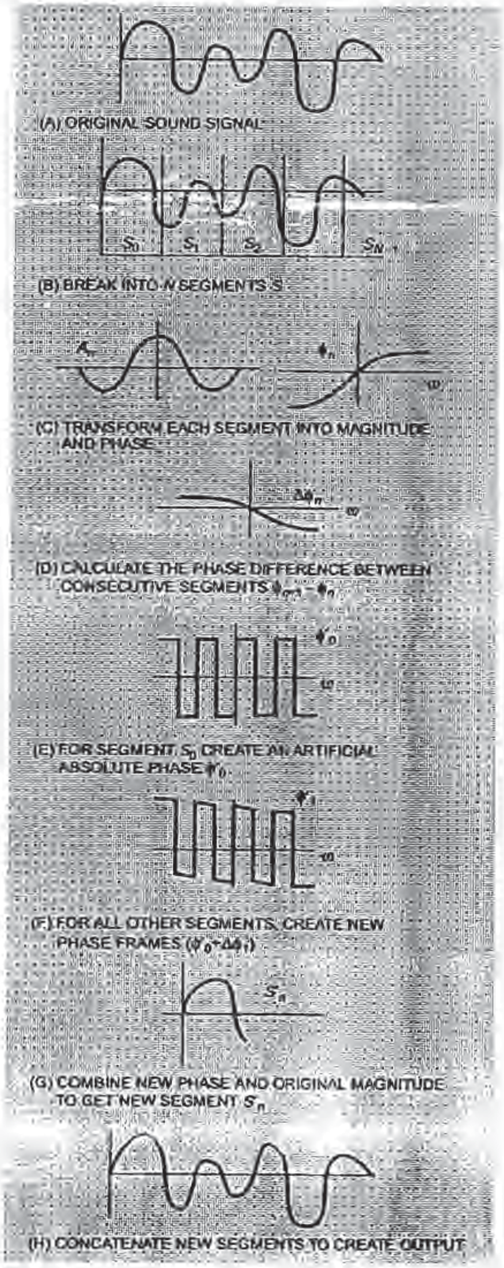
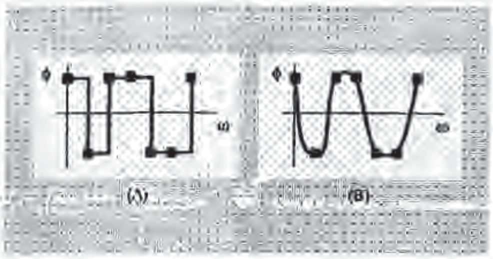


Figure 13 Sharp verses smooth transition



4. A binary set of data is represented as a $\phi_{data} = \pi/2$ or $-\pi/2$ representing 0 or 1 (Figure 12E):

$$\phi'_0 = \phi'_{data} \tag{14}$$

5. Re-create phase matrixes for $n > 0$ by using the phase difference (Figure 12F):

$$\begin{bmatrix} (\phi'_1(\omega_k) = \phi'_0(\omega_k) + \Delta\phi_1(\omega_k)) \\ \cdots \\ (\phi'_n(\omega_k) = \phi'_{n-1}(\omega_k) + \Delta\phi_n(\omega_k)) \\ \cdots \\ (\phi'_N(\omega_k) = \phi'_{N-1}(\omega_k) + \Delta\phi_N(\omega_k)) \end{bmatrix} \tag{15}$$

6. Use the modified phase matrix $\phi'_n(\omega_k)$ and the original magnitude matrix $A_n(\omega_k)$ to reconstruct the sound signal by applying the inverse DFT (Figure 12G, 12H).

For the decoding process, the synchronization of the sequence is done before the decoding. The length of the segment, the DFT points, and the data interval must be known at the receiver. The value of the underlying phase of the first segment is detected as a 0 or 1, which represents the coded binary string.

Since $\phi'_0(\omega_k)$ is modified, the absolute phases of the following segments are modified respectively. However, the relative [illegible] difference of each adjacent [illegible] is [illegible] this relative difference in phase that the [illegible] most sensitive to.

**Evaluation.** Phase dispersion is a distortion caused by a break in the relationship of the phases between each of the frequency components. Minimizing phase dispersion constrains the data rate of phase coding. One cause of phase dispersion is the substitution of phase $\phi'_0(\omega_k)$ with the binary code. The magnitude of the phase modifier needs to be close to the original value in order to minimize distortion. The difference between phase modifier states should be maximized in order to minimize the susceptibility of the encoding to noise. In our modified phase representation, a 0-bit is $-\pi/2$ and a 1-bit is $+\pi/2$.

Another source of distortion is the rate of change of the phase modifier. If distortion is applied to every bin of the DFT it is likely to break the phase relationship of the adjacent frequency components, resulting in a beat pattern. By changing the phase more slowly and transitioning between phase changes, the audible distortion is greatly reduced. In Figure 13, a sharp versus smooth transition is illustrated. In Figure 13A, the edges of the phase transitions are sharp, causing noticeable distortion. In Figure 13B, they are smooth, reducing this. Note that in each case, the data points appear in the same place. This smooth variation has the disadvantage of causing a reduction in bandwidth, as space has to be left between each data point to allow for smooth transition.

**Results.** In our experiments, the phase coding channel capacity typically varied from 8 bps to 32 bps, depending on the sound context. A channel capacity of ~8 bps can be achieved by allocating 128 frequency slots per bit under conditions of little background noise. Capacities of 16 bps to 32 bps can be achieved by allocating 32 to 64 frequency slots per slot when there is a noisy background.

## Spread spectrum

In a normal communication channel, it is often desirable to concentrate the information in as narrow a region of the frequency spectrum as possible in order to conserve available bandwidth and to reduce power. The basic spread spectrum technique, on the other hand, is designed to encode a stream of information by spreading the encoded data across as much of the frequency spectrum as possible. This allows the signal reception, even if there is interference on some frequencies.

While there are many variations on spread spectrum communication, we concentrated on Direct Sequence Spread Spectrum encoding (DSSS). The DSSS method spreads the signal by multiplying it by a *chip*, a maximal length pseudorandom sequence modulated at a known rate. Since the host signals are in discrete-time format, we can use the sampling rate as the *chip rate* for coding. The result is that the most difficult problem in DSSS receiving, that of establishing the correct start and end of the chip quanta for phase locking purposes, is taken care of by the discrete nature of the signal. Consequently, a much higher chip rate, and therefore a higher associated data rate, is possible. Without this, a variety of signal locking algorithms may be used, but these are computationally expensive.

**Procedure.** In DSSS, a *key* is needed to encode the information and the same key is needed to decode it. The key is pseudorandom noise that ideally has flat frequency response over the frequency range, i.e., white noise. The key is applied to the coded information to modulate the sequence into a spread spectrum sequence.

The DSSS method is as follows:[8,9] The code is multiplied by the carrier wave and the pseudorandom noise sequence, which has a wide frequency spectrum. As a consequence, the spectrum of the data is spread over the available band. Then, the spread data sequence is attenuated and added to the original file as additive random noise (see Figure 14). DSSS employs bi-phase shift keying since the phase of the signal alternates each time the modulated code alternates (see Figure 15). For decoding, phase values $\phi_0$ and $\phi_0 + \pi$ are interpreted as a "0" or a "1," which is a coded binary string.

In the decoding stage, the following is assumed:

1. The pseudorandom key is maximal (it has as many combinations as possible and does not repeat for as long as possible). Consequently it has a relatively flat frequency spectrum.
2. The key stream for the encoding is known by the receiver. Signal synchronization is done, and the start/stop point of the spread data are known.
3. The following parameters are known by the receiver: chip rate, data rate, and carrier frequency.

**Results.** Unlike phase coding, DSSS introduces additive random noise to the sound. To keep the noise level low and inaudible, the spread code is attenuated (without adding [illegible]) [illegible] dynamic [illegible] concatenation of simple repetition technique and error correction coding ensure the integrity of the code. A short segment of the binary code string is concatenated and added to the host signal so that transient noise can be reduced by averaging over the segment in the decoding stage. The resulting data rate of the DSSS experiments is 4 bps.

Figure 14 Spread spectrum encoding

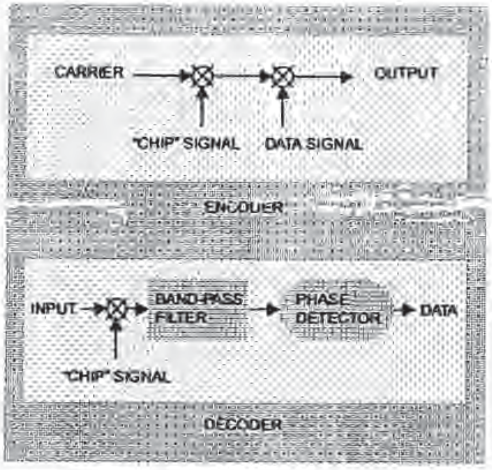
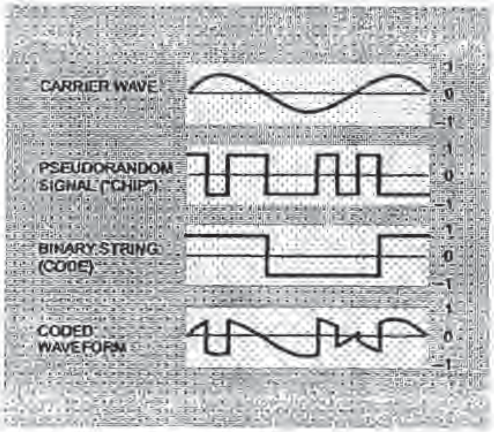


Figure 15 Synthesized spread spectrum information encoded by the direct sequence method



[illegible]

Echo data hiding embeds data into a host audio signal by introducing an *echo*. The data are hidden by varying three parameters of the echo: initial amplitude, decay rate, and offset (see Figure 16). As the offset (or

Figure 16 Adjustable parameters

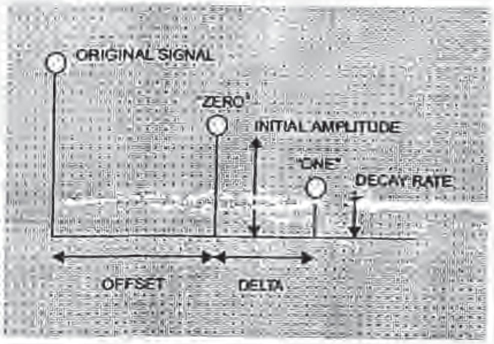


Figure 17 Discrete time exponential

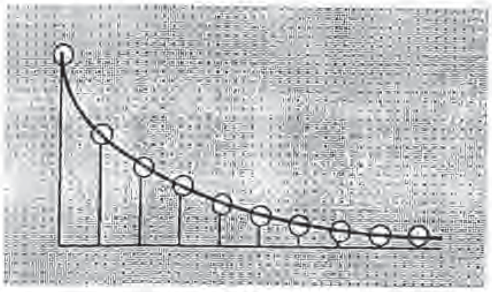
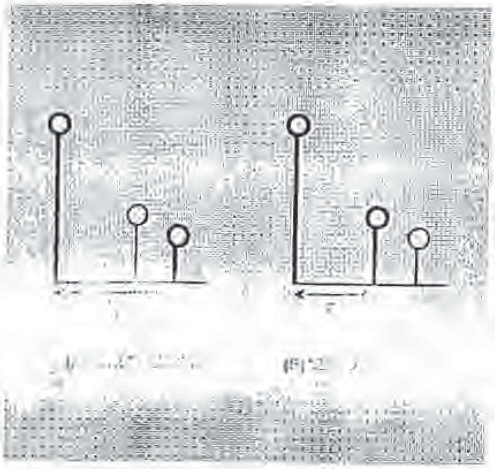


Figure 18 Echo kernels



delay) between the original and the echo decreases, the two signals blend. At a certain point, the human ear cannot distinguish between the two signals. The echo is perceived as added resonance. (This point is hard to determine exactly. It depends on the quality of the original recording, the type of sound being echoed, and the listener. In general, we find that this fusion occurs around 1/1000 of a second for most sounds and most listeners.)

The coder uses two delay times, one to represent a binary one (offset) and another to represent a binary zero (offset + delta). Both delay times are below the threshold at which the human ear can resolve the echo. In addition to decreasing the delay time, we can also ensure that the information is not perceivable by setting the initial amplitude and the decay rate below the audible threshold of the human ear.

**Encoding.** The encoding process can be represented as a system that has one of two possible system functions. In the time domain, the system functions are discrete time exponentials (see Figure 17) differing only in the delay between impulses.

For simplicity, we chose an example with only two impulses (one to copy the original signal and one to create an echo). Increasing the number of impulses is what increases the number of echoes.

We let the kernel shown in Figure 18A represent the system function for encoding a binary one and we use the system function defined in Figure 18B to encode a zero. Processing a signal through either Figures 18A or 18B will result in an encoded signal (see Figure 19).

The delay ($\delta_b$) between the original signal and the echo is dependent on which kernel or system function we use in Figure 19. The "one" kernel (Figure 18A) is created with a delay of ($\delta_1$) seconds while the "zero" kernel (Figure 18B) has a ($\delta_0$) second delay.

In order to encode more than one bit, the original signal is divided into smaller portions. Each individual portion can then be echoed with the desired bit by considering each as an independent signal. The final encoded signal (containing several bits) is the recombination of all independently encoded signal portions.

In Figure 20, the example signal has been divided into seven equal portions labeled a, b, c, d, e, f, and g. We want portions a, c, d, and g to contain a one. Therefore, we use the "one" kernel (Figure 18A) as the system function for each of these portions. Each portion is individually convolved with the system function. The zeros encoded into sections b, c, and f are encoded in a similar manner using the "zero" kernel (Figure 18B). Once each section has been individually convolved with the appropriate system function, the results are recombined. To achieve a less noticeable mix, we create a "one" echo signal by echoing the original signal using the "one" kernel. The "zero" kernel is used to create the "zero" echo signal. The resulting signals are shown in Figure 21.

The "one" echo signal and the "zero" echo signal contain only ones and zeros, respectively. In order to combine the two signals, two mixer signals (see Figure 22) are created. The mixer signals are either one or zero depending on the bit we would like to hide in that portion of the original signal.

The "one" mixer signal is multiplied by the "one" echo signal while the "zero" mixer signal is multiplied by "zero" echo signal. In other words, the echo signals are scaled by either 1 or 0 throughout the signal depending on what bit any particular portion is supposed to contain. Then the two results are added. Note that the "zero" mixer signal is the complement of the "one" mixer signal and that the transitions within each signal are ramps. The sum of the two mixer signals is always one. This gives us a smooth transition between portions encoded with different bits and prevents abrupt changes in the resonance of the final (mixed) signal.

A block diagram representing the entire encoding process is illustrated in Figure 23.

**Decoding.** Information is embedded into a signal by echoing the original signal with one of two delay kernels. A binary one is represented by an echo kernel with a ($\delta_1$) second delay. A binary zero is represented by a ($\delta_0$) second delay. Extraction of the embedded information involves the detection of spacing between the echoes. In order to do this, we examine the magnitude (at two locations) of the autocorrelation of the encoded signal's cepstrum:[14]

$$F^{-1}(\ln_{complex}(F(x))^2) \tag{16}$$

The following procedure is an example of the decoding process. We begin with a sample signal that is a series of impulses such that the impulses are separated by a set interval and have exponentially decaying

**Figure 19** Echoing example

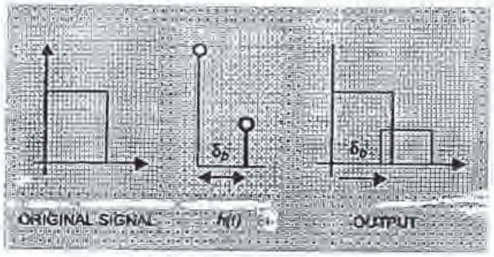


**Figure 20** Divide the original signal into smaller portions to encode information

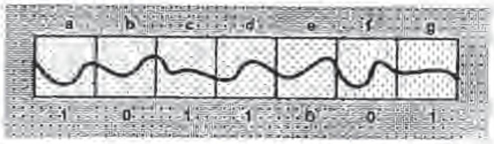


Figure 21 The first step in encoding process is to create a "one" and a "zero" echo signal (purple line is the echoed signal)

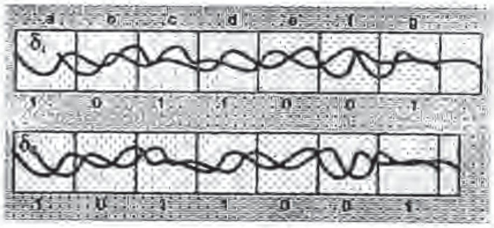


**Figure 22** Mixer signals

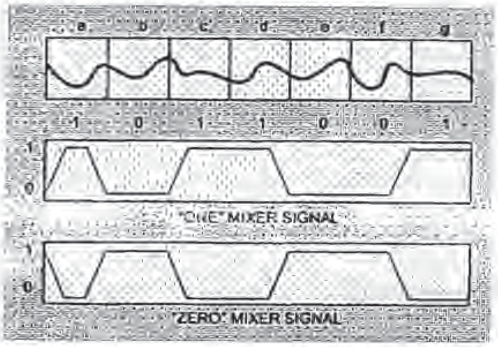


**Figure 23 Encoding process**

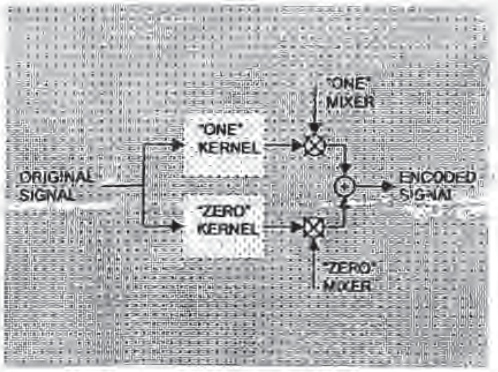
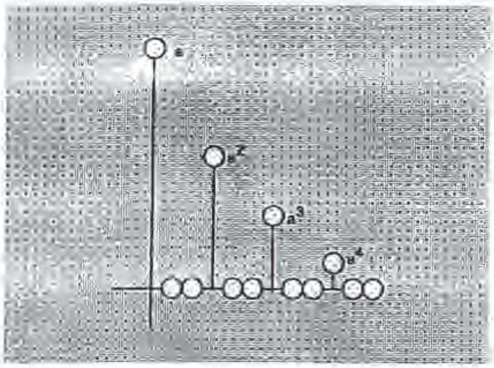


**Figure 24 Example signal: $x[n] = a^n u[n]$; $(0 < a < 1)$**

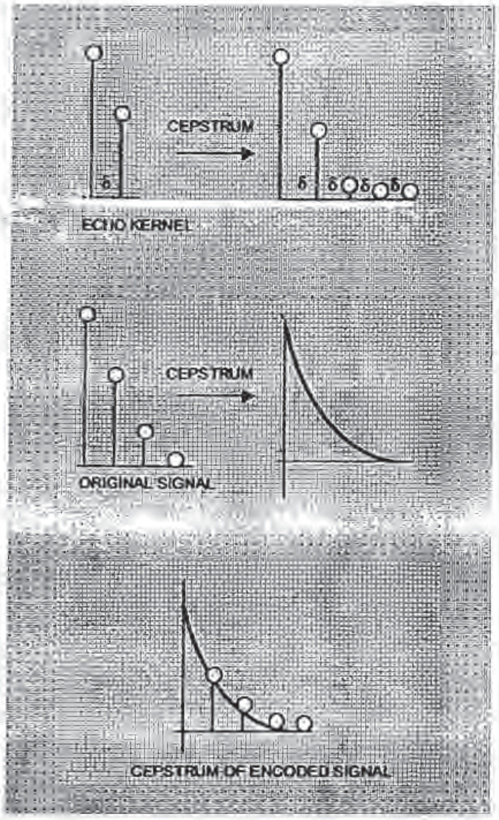


amplitudes. The signal is zero elsewhere (see Figure 24).

The next step is to find the cepstrum[14] of the echoed version. The result of taking the cepstrum makes the spacing between the echo and the original signal a little clearer.

Unfortunately, the result of the cepstrum also duplicates the echo every ($\delta$) seconds. In Figure 25, this is illustrated by the impulse train in the output. Furthermore, the magnitude of the impulses representing the echoes are small relative to the original signal. As such, they are difficult to detect. The solution to this problem is to take the autocorrelation of the cepstrum.

**Figure 25 Cepstrum of the echo-encoded signal**



We echo the signal once with delay ($\delta$) using the kernel depicted in Figure 26. The result is illustrated in Figure 27.

Only the first impulse is significantly amplified as it is reinforced by subsequent impulses. Therefore, we get a *spike* in the position of the first impulse. Like the first impulse, the spike is either ($\delta_1$) or ($\delta_0$) seconds after the original signal. The remainder of the impulses approach zero. Conveniently, random noise suffers the same fate as all the impulses after the first.

The rule for deciding on a one or a zero is based on the time delay between the original signal and the delay ($\delta$) before the spike in the autocorrelation.

Recall that a one was encoded by placing an echo ($\delta_1$) seconds after the original and a zero was placed ($\delta_0$) seconds after the original. When decoding, we assign a one if the magnitude of the autocorrelation function is greater at ($\delta_1$) seconds than it is at ($\delta_0$) seconds. A zero is assigned if the reverse is true. This is the same as deciding which kernel we used utilizing the fact that the "one" and "zero" kernel differ only in the delay before the echo (Figure 18).

**Results.** Using the methods described, it is indeed possible to encode and decode information in the form of binary digits into a media stream with minimal alteration to the original signal at approximately 16 bps (see Figure 28). By minimal alteration, we mean that the output of the encoding process is changed in such a way so that the average human cannot hear any significant difference between the altered and the original signal. There is little, if any, degradation of the original signal. Instead, the addition of resonance simply gives the signal a slightly richer sound. While the addition of resonance may be problematic in some music applications, studio engineers may be able to fine-tune the echo hiding parameters during the mastering process, enabling its use.

## Supplemental techniques

Three supplemental techniques are discussed next.

**Adaptive data attenuation.** The optimum attenuation factor varies as the noise level of the host sound changes. By adapting the attenuation to the short-term changes of the sound or noise level, we can keep the coded noise extremely low during the silent segments and increase the coded noise during the noisy segments. In our experiments, the quantized magnitude envelope of the host sound wave is used as a reference value for the adaptive attenuation; and the maximum noise level is set to 2 percent of the dynamic range of the host signal.

**Redundancy and error correction coding.** In order to compensate for errors due to channel noise and host signal modification, it is useful to apply error-correction coding (ECC)[1] to the data to be embedded. While there exist some efficient methods of ECC, its application always results in a trade-off between robustness and data rate.

**Sound context analysis.** The detectability of white noise inserted into a host audio signal is linearly dependent upon the original noise level of the host signal. To maximize the quantity of embedded data, while ensuring the data are unnoticed, it is useful to express the noise level quantitatively. The noise level is characterized by computing the magnitude of change in adjacent samples of the host signal:

$$\sigma^2_{local} = \frac{1}{|S_{max}|} \times \frac{1}{N} \times \sum_{n=1}^{N-1} [s(n+1) - s(n)]^2 \qquad (17)$$

Figure 26 Echo kernel used in example

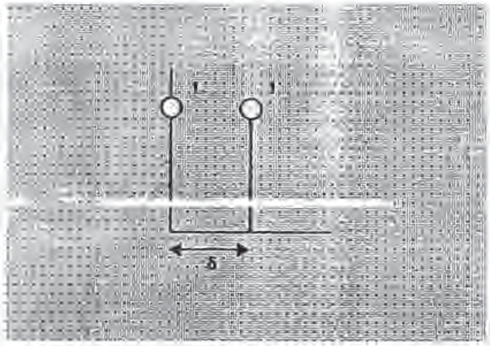


Figure 27 Echoed version of the example signal

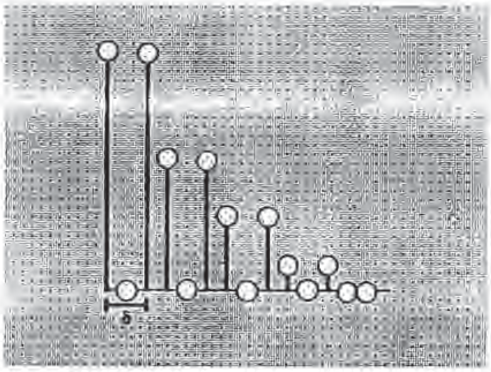
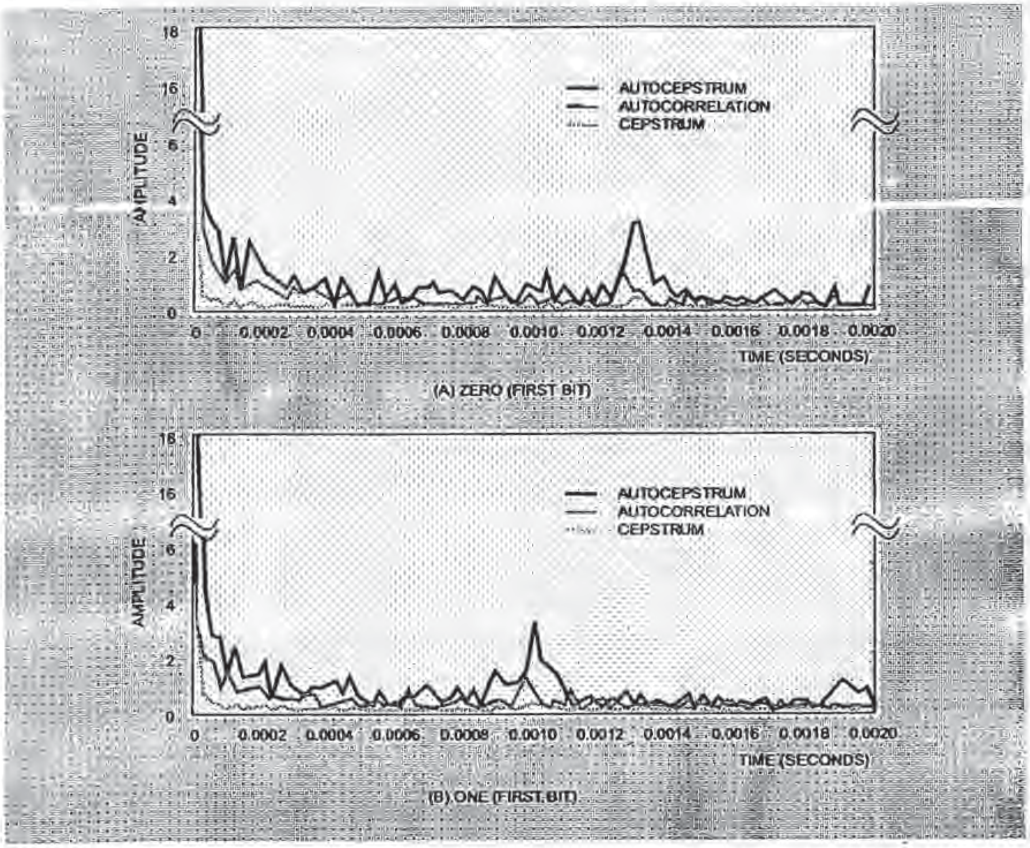


Table 2 Audio noise level analysis

| $\sigma^2_{local}$ | Quality |
|---|---|
| < 0.005 | Studio |
| > 0.01 | Crowd noise |

**Figure 28** Result of autocepstrum and autocorrelation for (A) "zero" and (B) "one" bits



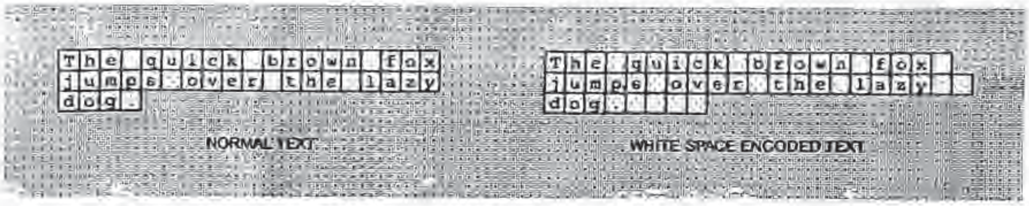
where $N$ is the number of sample points in the sequence and $S_{max}$ is the maximum magnitude in the sequence. We use this measure to categorize host audio signals by noise level (see Table 2).

## Data hiding in text

Soft-copy text is in many ways the most difficult place to hide data. (Hard-copy text can be treated as a highly structured image and is readily amenable to a variety of techniques such as slight variations in letter forms, kerning, baseline, etc.) This is due largely to the relative lack of redundant information in a text file as compared with a picture or a sound bite. While it is often possible to make imperceptible modifications to a picture, even an extra letter or period in text may be noticed by a casual reader. Data hiding in text is an exercise in the discovery of modifications that are not noticed by readers. We considered three major methods of encoding data: open space methods that encode through manipulation of white space (unused space on the printed page), syntactic methods that utilize punctuation, and semantic methods that encode using manipulation of the words themselves.

**Open space methods.** There are two reasons why the manipulation of white space in particular yields useful results. First, changing the number of trailing spaces has little chance of changing the meaning of a phrase or sentence. Second, a casual reader is unlikely to take

**Figure 29** Example of data hidden using white space



notice of slight modifications to white space. We describe three methods of using white space to encode data. The methods exploit inter-sentence spacing, end-of-line spaces, and inter-word spacing in justified text.

The first method encodes a binary message into a text by placing either one or two spaces after each terminating character, e.g., a period for English prose, a semicolon for C-code, etc. A single space encodes a "0," while two spaces encode a "1." This method has a number of inherent problems. It is inefficient, requiring a great deal of text to encode a very few bits. (One bit per sentence equates to a data rate of approximately one bit per 160 bytes assuming sentences are on average two 80-character lines of text.) Its ability to encode depends on the structure of the text. (Some text, such as free-verse poetry, lacks consistent or well-defined termination characters.) Many word processors automatically set the number of spaces after periods to one or two characters. Finally, inconsistent use of white space is not transparent.

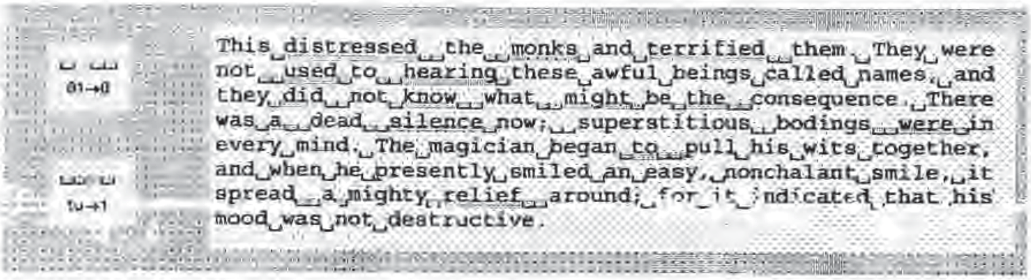
A second method of exploiting white space to encode data is to insert spaces at the end of lines. The data are encoded allowing for a predetermined number of spaces at the end of each line (see Figure 29). Two spaces encode one bit per line, four encode two, eight encode three, etc., dramatically increasing the amount of information we can encode over the previous method. In Figure 29, the text has been selectively justified, and has then had spaces added to the end of lines to encode more data. Rules have been added to reveal the white space at the end of lines. Additional advantages of this method are that it can be done with any text, and it will go unnoticed by readers, since this additional white space is peripheral to the text. As with the previous method, some programs, e.g., "sendmail," may inadvertently remove the extra space characters. A problem unique to this method is that the hidden data cannot be retrieved from hard copy.

A third method of using white space to encode data involves right-justification of text. Data are encoded by controlling where the extra spaces are placed. One space between words is interpreted as a "0." Two spaces are interpreted as a "1." This method results in several bits encoded on each line (see Figure 30). Because of constraints upon justification, not every inter-word space can be used as data. In order to determine which of the inter-word spaces represent hidden data bits and which are part of the original text, we have employed a Manchester-like encoding method. Manchester encoding groups bits in sets of two, interpreting "01" as a "1" and "10" as a "0." The bit strings "00" and "11" are null. For example, the encoded message "1000101101" is reduced to "001," while "110011" is a null string.

Open space methods are useful as long as the text remains in an ASCII (American Standard Character Interchange) format. As mentioned above, some data may be lost when the text is printed. Printed documents present opportunities for data hiding far beyond the capability of an ASCII text file. Data hiding in hard copy is accomplished by making slight variations in word and letter spacing, changes to the baseline position of letters or punctuation, changes to the letter forms themselves, etc. Also, image data-hiding techniques such as those used by Patchwork can be modified to work with printed text.

**Syntactic methods.** That white space is considered arbitrary is both its strength and its weakness where data hiding is concerned. While the reader may not notice its manipulation, a word processor may inadvertently change the number of spaces, destroying the hidden data. Robustness, in light of document reformatting, is one reason to look for other methods of data hiding in text. In addition, the use of syntactic and semantic methods generally does not interfere with the open space methods. These methods can be applied in parallel.

**Figure 30** Data hidden through justification (text from *A Connecticut Yankee in King Arthur's Court* by Mark Twain)



```
This␣distressed␣␣the␣␣monks␣and␣terrified␣␣them.␣They␣were
not␣␣used␣to␣␣hearing␣these␣awful␣beings␣called␣names,␣␣and
they␣did␣␣not␣know␣␣what␣␣might␣be␣the␣␣consequence.␣There
was␣a␣␣dead␣␣silence␣now;␣␣superstitious␣␣bodings␣␣were␣in
every␣mind.␣The␣magician␣began␣to␣␣pull␣his␣wits␣together,
and␣when␣he␣presently␣smiled␣an␣easy,␣nonchalant␣smile,␣it
spread␣␣a␣mighty␣relief␣␣around;␣for␣it␣indicated␣that␣his
mood␣was␣not␣destructive.
```

**Table 3** Synonymous pairs

| | | |
|---|---|---|
| big | ≈ | large |
| small | ≈ | little |
| chilly | ≈ | cool |
| smart | ≈ | clever |
| spaced | ≈ | stretched |

There are many circumstances where punctuation is ambiguous or when mispunctuation has low impact on the meaning of the text. For example, the phrases "bread, butter, and milk" and "bread, butter and milk" are both considered correct usage of commas in a list. We can exploit the fact that the choice of form is arbitrary. Alternation between forms can represent binary data, e.g., anytime the first phrase structure (characterized by a comma appearing before the "and") occurs, a "1" is inferred, and anytime the second phrase structure is found, a "0" is inferred. Other examples include the controlled use of contractions and abbreviations. While written English affords numerous cases for the application of syntactic data hiding, these situations occur infrequently in typical prose. The expected data rate of these methods is on the order of only several bits per kilobyte of text.

Although many of the rules of punctuation are ambiguous or redundant, inconsistent use of punctuation is noticeable to even casual readers. Finally, there are cases where changing the punctuation will impact the clarity, or even meaning, of the text considerably. This method should be used with caution.

Syntactic methods include changing the diction and structure of text without significantly altering meaning or tone. For example, the sentence "Before the night is over, I will have finished" could be stated "I will have finished before the night is over." These methods are more transparent than the punctuation methods, but the opportunity to exploit them is limited.

**Semantic methods.** A final category of data hiding in text involves changing the words themselves. Semantic methods are similar to the syntactic method. Rather than encoding binary data by exploiting ambiguity of form, these methods assign two synonyms primary or secondary value. For example, the word "big" could be considered primary and "large" secondary. Whether a word has primary or secondary value bears no relevance to how often it will be used, but, when decoding, primary words will be read as ones, secondary words as zeros (see Table 3).

Word *webs* such as WordNet can be used to automatically generate synonym tables. Where there are many synonyms, more than one bit can be encoded per substitution. (The choice between "propensity," "predilection," "penchant," and "proclivity" represents two bits of data.) Problems occur when the nuances of meaning interfere with the desire to encode data. For example, there is a problem with choice of the synonym pair "cool" and "chilly." Calling someone "cool" has very different connotations than calling them "chilly." The sentence "The students in line for registration are spaced-out" is also ambiguous.

## Applications

Data hidden in text has a variety of applications, including copyright verification, authentication, and annotation. Making copyright information inseparable from the text is one way for publishers to protect their products in an era of increasing electronic distribution. Annotation can be used for tamper protection. For example, if a cryptographic hash of the paper is encoded into the paper, it is a simple matter to determine whether or not the file has been changed. Verification is among the tasks that could easily be performed by a server which, in this case, would return the judgment "authentic" or "unauthentic" as appropriate.

Other uses of data hiding in text involve embedding instructions for an autonomous program in a text. For example, a mail server can be programmed to check for hidden messages when transmitting an electronic message. The message is rejected or approved depending on whether or not any hidden data are found. In this way a company running its own mail server can keep confidential documents from being inadvertently exported.

## Conclusion

In this paper, several techniques are discussed as possible methods for embedding data in host text, image, and audio signals. While we have had some degree of success, all of the proposed methods have limitations. The goal of achieving protection of large amounts of embedded data against intentional attempts at removal may be unobtainable.

Automatic detection of geometric and nongeometric modifications applied to the host signal after data hiding is a key data-hiding technology. The optimum trade-offs between bit rate, robustness, and perceivability need to be defined experimentally. The interaction between various data-hiding technologies needs to be better understood.

While compression of image and audio content continues to reduce the necessary bandwidth associated with image and audio content, the need for a better contextual description of that content is increasing. Despite its current shortcomings, data-hiding technology is important as a carrier of these descriptions.

## Acknowledgments

This work was supported in part by the News in the Future research consortium at the MIT Media Laboratory and International Business Machines Corporation.

## Cited references

1. P. Sweene, *Error Control Coding (An Introduction)*, Prentice-Hall International Ltd., Englewood Cliffs, NJ (1991).
2. E. Adelson, *Digital Signal Encoding and Decoding Apparatus*, U.S. Patent No. 4,939,515 (1990).
3. R. Machado, "Stego," http://www.nitv.net/~mech/Romana/stego.html (1994).
4. W. Bender, "Data Hiding," News in the Future, MIT Media Laboratory, unpublished lecture notes (1994).
5. A. Lippman, *Receiver-Compatible Enhanced EDTV System*, U.S. Patent No. 5,010,405 (1991).
6. D. L. Hecht, "Embedded Data Glyph Technology for Hardcopy Digital Documents," *SPIE* 2171 (1994).
7. K. Matsui and K. Tanaka, "Video-Steganography: How to Secretly Embed a Signature in a Picture," *IMA Intellectual Property Project Proceedings* (1994).
8. R. C. Dixon, *Spread Spectrum Systems*, John Wiley & Sons, Inc., New York (1976).
9. S. K. Marvin, *Spread Spectrum Handbook*, McGraw-Hill, Inc., New York (1985).
10. Digimarc Corporation, *Identification/Authentication Coding Method and Apparatus*, U.S. Patent (1995).
11. I. Cox, J. Kilian, T. Leighton, and T. Shamoon, "Secure Spread Spectrum Watermarking for Multimedia," *NECI Technical Report* 95-10, NEC Research Institute, Princeton, NJ (1995).
12. A. V. Drake, *Fundamentals of Applied Probability*, McGraw-Hill, Inc., New York (1967).
13. L. R. Rabiner and R. W. Schaffer, *Digital Processing of Speech Signal*, Prentice-Hall, Inc., Englewood Cliffs, NJ (1975).
14. A. V. Oppenheim and R. W. Shaffer, *Discrete-Time Signal Processing*, Prentice-Hall, Inc., Englewood Cliffs, NJ (1989).

*Accepted for publication February 29, 1996.*

**Walter Bender** *MIT Media Laboratory, 20 Ames Street, Cambridge, Massachusetts 02139-4307 (electronic mail: walter@media.mit.edu).* Mr. Bender is a principal research scientist at the MIT Media Laboratory and principal investigator of the laboratory's News in the Future consortium. He received the B.A. degree from Harvard University in 1977 and joined the Architecture Machine Group at MIT in 1978. He received the M.S. degree from MIT in 1980. Mr. Bender is a founding member of the Media Laboratory.

**Daniel Gruhl** *MIT Media Laboratory, 20 Ames Street, Cambridge, Massachusetts 02139-4307 (electronic mail: druid@mit.edu).* Mr. Gruhl is a doctoral student in the department of electrical engineering and computer science at MIT, where he earned an S.B. in 1994 and an MEng in 1995. He is an AT&T fellow and research assistant at the Media Lab, where his research interests include information hiding in images, sound, and text, as well as user modeling for electronic information systems.

**Norishige Morimoto** *IBM Tokyo Research Laboratory, 1623-14 Shimo-Tsuruma, Yamato-shi, Kanagawa-ken, 242 Japan (electronic mail: nly@trl.ibm.co.jp).* Mr. Morimoto is currently a researcher at the Tokyo Research Laboratory (TRL) of IBM Japan. He joined IBM in 1987 and was transfered to the TRL in 1995 after getting his master's degree from MIT. His area of interest is rights management of digital contents and network applications for industry solutions. Mr. Morimoto received his B.S. degree in electrical engineering from Keio University, and his M.S. degree in electrical engineering and computer science from the Massachusetts Institute of Technology.

**Anthony Lu** *MIT Media Laboratory, 20 Ames Street, Cambridge, Massachusetts 02139-4307 (electronic mail: tonyl@media.mit.edu).* Mr. Lu is an undergraduate student in electrical engineering at MIT, concentrating in communications and signal processing. He will be a candidate for the master of engineering degree in 1997.

Reprint Order No. G321-5608.

## This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

## BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

☐ BLACK BORDERS

☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES

☐ FADED TEXT OR DRAWING

☑ BLURRED OR ILLEGIBLE TEXT OR DRAWING

☐ SKEWED/SLANTED IMAGES

☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS

☑ GRAY SCALE DOCUMENTS

☐ LINES OR MARKS ON ORIGINAL DOCUMENT

☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY

☐ OTHER: ______________________________

**IMAGES ARE BEST AVAILABLE COPY.**
**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**

## Visions and Views

**Editor: Nevenka Dimitrova**
***Philips Research***

# Bandwidth as Currency

**Scott Moskowitz**
***Blue Spike***

While cash and its equivalents let us objectify commercial transactions, *money*, a medium of exchange, is simply information.[1] Provisioning bandwidth in an actuarially consistent manner has many far-reaching implications reminiscent of the commercial rationalization of railways in the 1870s—in that period's *path to profitability*. Mapping packet flow to willingness to pay, as rail cargo content was matched to the cost of carry, will better utilize bandwidth, enhancing network value consistent with the information being exchanged. The notion of a *packet watermark* can enable bandwidth as currency. Payment facilities *are generally* treated as separate, independent data flows from the actual data being transacted. Packet watermarks map payment facilities to the fidelity, discreteness, or functionality of the data demanded, representing a consistent means of determining a willingness to pay. This mapping acts as a receipt for data commerce. Uniquely identifying the exchange of objects representing *abstractions of value*[2] enhances a transparent, liquid information economy.[3,4]

The basic framework of areas that would enable this bandwidth provisioning includes

- efficient packet provisioning on a network using a packet watermark,
- unique identification of bandwidth availability and flow,
- bandwidth credentials creation to enhance liquidity and derivative pricing for future estimated use of bandwidth, and
- cryptographic protocol-based rules that let market mechanisms objectively bill and subsequently resolve disputes.[5]

Current information commerce of media or functionally rich data objects typically lacks any assessment of responsibility for the parties and intermediaries handling data objects. Applying cryptographic uniqueness to the packets and traditional cryptographic key-based watermarking to the data objects would uniquely identify the object and monitor the information exchange.[6]

Packet watermarks differ from traditional digital watermarks. Whereas digital watermarks act on the application-layer data object intrinsically related to type and use of the data,[7,8] packet watermarks relate to the actual transmission. Packet watermarks assist with the authenticated provisioning of packet flows between users, can break the actual transmission into parts, resequence the parts, and introduce additional communication-related information—which can later be associated with each other. Preferably, the packet watermarked data won't interfere with the traditional digital watermarks, which establish responsibility for the objects being transacted.

As with other transmissions, end users don't care about the nature of the packets. However, they can benefit from using the best paths for getting information. Vendors offering information could use packet-watermarking applications to objectively assess responsibility for data—for legal or economic reasons. This could also avoid double payments of bandwidth, where vendors handle the sending and receiving costs, instead of an optimized path between a sender and a receiver. These applications could enhance trust in entities (that is, devices and people) that are increasingly associated with some intangible, yet recognizable information associated with the transaction itself. *Trusted computing* might result from these approaches.

### Tragedy of the network commons

Bandwidth suffers from the tragedy of the commons, that is, everyone wants to increase their quota, even if it's detrimental to the global system. We need transparent, liquid, and secure

1070-986X/03/$17.00 © 2003 IEEE

Published by the IEEE Computer Society

protocols to accurately assess and provision bandwidth in a manner consistent with responsible information commerce. Some refer to this issue from a public policy perspective as *spectrum management*, others as *the future of ideas*.[9]

Addressing the optimized allocation of bandwidth has largely been the domain of quality of service (QoS) approaches, competitively offered in tandem with traditional peering arrangements between large carriers. Early work resulted in caching technologies, which push higher demand data closer to the access points for which the data is demanded. QoS attempts to make decisions about bandwidth accessibility based on a user's ability to access information within some predetermined time frame. For instance, if $X$ number of users can access $Y$ amount of bandwidth over some fixed period of time $T$, we can estimate bandwidth as a function of satisfying users $X$, or some percentage of $X$, for each increment of $Y$ divided by $T$.

Traditional telephone billing systems provide a somewhat accurate measure of bandwidth use, measured as discrete instants of time, and the general, or hybridized, path by which users are connected. However, present information commerce of media or functionally rich data objects typically lack any assessment of responsibility for the parties and intermediaries handling these objects.[10] Blue Spike has developed a number of novel concepts based on 10 years of research and development in intellectual property rights management and cryptographic payment systems.

While priority of transmission paths helps alleviate bottlenecks within a given network, mapping demand for bandwidth has become increasingly difficult. This might result from user's assigning a high priority to their data. It also could be a result of competing interests within the Internet service provider space.

Several technological approaches to the bottleneck issues attempt to minimize computational overhead. Data compression schemes for media-rich content that support streaming or sharing an audio or video signal—for example MPEG-4—reduce the total number of bits transmitted over the communication channel. In the functional data space, optimized languages attempt to reduce computational overhead for a variety of applications, including multimedia messaging services or Java midlets. The reality is, not all data in a particular format or market segment carries equal commercial value. The market's participants have a deficit of time to offer and enter into commercial transactions, placing a premium on accessibility and satisfaction of good or service demands close to real time. Networks should let market mechanisms assist in providing and pricing data that's consistent with the bandwidth requirements and the rights of content or software creators while not interfering with the consumer's experience in transacting data.[11]

Internet protocol (IP) provides each networked device with an IP address. IP version 4 (IPv4) incorporates option fields that can be exploited at any place in the transmission chain for writing/embedding and detecting/recovering a specialized type of digital watermark that's suited for provisioning and pricing schemes, bandwidth prioritization, management systems, and dispute resolution and clearinghouse functions. Because of the sequential nature of TCP/IP, network researchers have suggested assigning higher priority to the perceptibly significant data in a data object.

## Nonsequential transport for bandwidth provisioning

One way to optimize data transmission speed is based on Reed–Solomon error-correction coding. TCP/IP packets represent predetermined packets of data, that is, they have a specific size without regard to the data object being rendered. Therefore, coarser estimates of the data objects' aesthetics or signal characteristics let mathematical values be assigned to a larger portion or subset of the data object. A simple linear equation can define the independently derived values representing the data object. These mathematical values represent groupings of packets that aren't sequentially ordered but fitted to the characteristics of the data object being broken down for transmission. Additionally, systems or devices related to sending and receiving data can handle these values to speed data transmission.

Data chunks aren't sequential with error-correction coding, as it is with TCP, but are generated with variations on the Reed–Solomon code. As a result, receivers of the data get transmission chunks that can be reconstructed nonsequentially, but efficiently, so long as they receive assigned data values. The chunks may also overlap the packets that would typically represent the object. On the receiving end of the transmission, some applications first reconstruct those data signal features deemed perceptibly significant. Medical data, which might be time sensitive, can benefit from this form of transmission. This approach speeds the routing of data over a network in a manner consistent with the perceptible value of the data, but it still lacks an effective way of attributing responsibility over data transmissions.

### Tiered bandwidth quality with traditional digital watermarks

A wholly different approach combines traditional digital watermarks embedded in a full-bandwidth signal. These signals might have distortions or quality levels intentionally introduced that have differential pricing levels associated with predetermined keys for formulating a subset of the original signal's quality level and a rough estimate of overall signal quality demand (via the exchange of authentication information carried by embedded signals in the streamed data objects).[12] Each client would still receive a full-bandwidth signal at some level of quality up to full, and a yield in time measured via the verification of the embedded bits reported back to the server.[13]

Using transfer functions—which weigh the input to output of data—introduces degraded quality levels as a form of *chaffing* or *scrambling*. An approach that has a relationship with the signal's characteristics would not require separately handling and encrypting each quality level of a given signal served on a per request basis. Here, I discuss higher bandwidth granularity in observing the link between information, quality, and demand.

### Business side of bandwidth provisioning

IPv6 includes proposals for additional optimizations. In contrast with current IPv4 systems optimized to handle end-to-end data transmission without regard for the data's content, IPv6 will enable traffic prioritization, low-level authentication with encryption, and better handling of audio and video streams. The labeling scheme discussed in this article enables better granularity in handling data packets with a labeling scheme over network infrastructures. The approach's authentication protocol prevents labeling fraud to reduce freeloading on paid bandwidth flows. The method uses packet flow watermarks differently than traditional digital watermarking. It prioritizes data traffic and defines the transmitted data so that it's consistent with the rights of the content or the data's functionality. The method also includes provisions for clearinghouse facilities and certification of traffic. Further, it offers secondary or derivative markets for assisting in efficient pricing of future bandwidth. From these novel techniques, I anticipate appropriate digital credentials for bandwidth pricing and use—called a *bandwidth credential* or *bandwidth digital certificate* as per traditional cryptological terminology.

We can now address market-based pricing of data in a manner that provides bandwidth efficiently. When a steganographic cipher or cryptographic-key based method watermarks a single data object, aesthetic or functional, it can be made unique.[14,15] Uniquely watermarking flows of packets, postage for packets (bandwidth provisioning) represents a natural extension for mapping granular commercial value of demanded packets versus other packets. By associating identifying and authenticating information of the watermark flows of packets, networks can more efficiently apportion bandwidth to meet market demands. The steps of identification, authentication, verification, and authorization are like negotiable levels of information exchange required by either party to a transaction. Certain types of transactions will require more or less information exchange than others, including higher security protocol demands to flexibly handle as many potential transactions as possible and bit commitments—as with zero knowledge signature schemes—by one of more of the parties for any additional assurance. More specifically, demand for information over networks and a better ability to identify the packets people are willing to pay for can be enabled in a highly efficient, cost-effective manner when demand is mapped to packets and their paths.

What also results is a better accounting system that provides billing packets to the appropriate parties and resolves disputes more objectively because cryptographic protocols assure a higher level of confidence in how provisioning is handled. Similarly, packet watermarking makes it possible to charge for bandwidth so that it resembles traditional telephone billing systems, albeit based on the value of data objects and the demands for the underlying packets in terms of time, quality, or functionality. The difference is that telephone billing systems don't consider the contents or paths of packets, nor do traditional telephone systems assist in creating a means for competitively evaluating bandwidth based on consumer demand for data. This demand can be compared to a more consistent media or in functional terms (type of media, associated rights, authenticity of the data, quality level of the media based on a differential price, optimized functions, code or algorithms, and so on) and not solely on data size terms.

A network, thus enabled, can check and verify efficient bandwidth delivery on a packet level and can store information concerning better paths between senders and receivers. For certain economic or business models, further features can be added to make Internet *handling* of data similar to how billing works for traditional telecommunications companies. Such companies buy bandwidth resources in bulk and don't necessarily have any underlying understanding of what the bandwidth is used for, why it's being demanded, nor how to encourage higher value-added for any given bit for each bit per time calculation. The following describes one framework for measuring bandwidth:

- The intrinsic value $V_I = X \times (\min_0 - \min_1)$, is the money saved in telecommunications costs by using a higher bandwidth. The intrinsic value can be negative, implying a compensating premium placed on the time saved by using a more expensive transport. Note that $\min_0 \geq \min_1$.
- The percentage chance of failure represents the chance a user can't exercise rights (immediate purchase or sale of bandwidth) or option (where the option is the right, but not obligation to purchase the underlying asset) for bandwidth. If the probability of failure is $P_f$, where $0 \leq P_f \leq 1$, and the value of the right is $V_0$, in the absence of failure, then $V_f = (1 - P_f)V_0$.
- The convenience premium might apply to the particular or uniquely identifiable data objects, whether the data object is streamed, date or time schedules, geographic locations of either the provider or user, the hardware or software underlying the network, or some other unique circumstances including live performances. The more demand in excess of supply, the higher convenience $C$, will rise. $V_C$ is then a function of supply and demand. Thus, $V_{real} = V_{theoretical} + V_C$.
- The time value is a function of the exercise period of a bandwidth right. It's proportional to $P_f$, since more time allows for transfer of recovery from an individual failure. There are two components of time: over what period a transfer can be initiated, and for how long the transfer can last once initiated. Thus, overall, $V = (1 - P_f)(V_I + V_T + V_C) = (1 - P_f)[(X(\min_0 - \min_1)) + V_T + V_C)]$ (Convenience premium $V_C$ should be independent of all other values, except $V$.)

The pricing model also incorporates classic Black–Scholes options pricing, or derivations of this model, to price future value for bandwidth.[16] The following properties describe Black–Scholes: The standard deviation of the asset's value (in this case, bandwidth, or that which is optioned) multiplied by the square root of the time of the option's expiration. Essentially a ratio of the asset value to the present value of the option's strike price represents the underlying property of future price. The strike price is the price at which the option is offered and later exercised. To purchase or to sell is the difference in the right of the option and is called a *call* or a *put* (a put is the right, but not obligation to sell; a call is the right but not obligation to buy the underlying asset). More generally, the Black-Scholes equation is as follows:

$C_0 = S_0\ N(d_1) - X\ e^{-r_f T}\ N(d2)$
Where
$S_0$ = the price of the underlying asset (a predetermined value)
$N(d_1)$ = the cumulative normal probability of unit normal variable $d_1$
$N(d_2)$ = the cumulative normal probability of unit normal variable $d_2$
$X$ = the exercise price
$T$ = the time to expiration or maturity of the option
$rf$ = the risk free rate (a value that can be predetermined at the time of pricing the option)
$e$ = the base of natural logarithms, constant = 2.7128…
$d_1 = [(ln\ (S / X) + r_f\ T) / (S \div T)] + [1/(2\ S \div T)]$
$d_2 = d_1 - S \div T$

Because the denominator (time) is fixed at any discrete moment, thus maximizing the economic value for the numerator (the bit) given a market for information goods and services, a higher economic value can be attributed to a given network that implement the features I describe here. While no one can know in advance the demand for a given data object, parties can agree to the cost of bandwidth for a given business activity (such as streaming a live concert or handling bandwidth-based transactions tied to a subscription with a bandwidth device such as a cell phone). Streaming, to date, isn't economically viable because vendors haven't taken a packet level view of the flow of data to people demanding a stream. Nor have vendors tied payment or willingness to pay to the packets in a consistent manner with the data being consumed.

Ultimately, the notions presented in this article emphasize the different needs of providers and consumers of content and the multivalent nature of trust. So long as some preexisting payment or credit facility exists, decisions or policies regarding the level and detail of security or credentials should be made as flexible as possible regarding data and computational resources. Some transactions might only require a check sum not a more secure, independently verifiable cryptographic digital credential such as a digital signature with an ITU-T X.509 digital certificate. Combining verifiable identification inherent to digital certificates with bandwidth provisioning results in a bandwidth digital credential. For networked devices, payment facilities can easily be enabled and tightly integrated, especially if such devices have IP addresses or some similar uniquely attributable ID.

We can enhance *tangible products* with the unique information and transaction processing as a basis for serializing the actual article of manufacture. A major thesis of the techniques described here is that commerce must balance privacy with concerns about piracy. Further, commerce is about uniqueness or the receipts for copies sold, not originals, for which uniqueness may be nonverifiable. Recognition, not physical location, begets the commercial need for establishing responsibility over copies, whether aesthetic or functional data.

## Packet watermarking

When a receiver requests a data object from a sender, the sender creates a packet flow with the receiver's address and sends it to the Internet. The packets might make many hops *in the cloud* before arriving at the receiver's IP address. At each node, a router examines the address and chooses a route to the next node. Often, there are many possible routes from each node to the final destination. These routes might be ranked by a number of criteria, including current load, historical load and reliability, and current and historical latency. All these factors could help route individual packets by more or less optimal paths—assuming that the router could discriminate between different flows. The packet watermark becomes the method by which the router identifies streams and creates differential QoS.

A packet watermark is cryptographically associated with the contents of the packet itself. An important issue is that the packets might contain functional data as opposed to aesthetic data. Mapping demand via cryptographic protocols to aesthetic data—in perceptibly significant portions of a signal—is only slightly different from mapping functionally significant data such as source, object, or executable code. For example, a traditional digital watermark might depend on the signal characteristics of the signal being watermarked. If watermarking occurs within a key-based system, a cryptographic association between the key and the signal or function via the watermark might exist. Besides the noise or signal characteristics in the signal, the key can be seeded by independent, random information to make it more difficult to decode, even if a potential pirate found the watermark in the signal.

Benefits from key-based watermarks are multifold. Key-based watermarks verify a data signal or object to establish responsibility for the signal or alert users of unauthorized copies. Similarly, a packet watermark sniffer could detect unauthorized use of a particular routing priority. The sniffer samples a fraction of the overall traffic to detect, and deter, abuse of the system. It reads the watermark on the packet, checks the authentication, and signals invalid packets. If necessary, the flow can then be rerouted or halted, depending on the terms of the commercial contract. Additional benefits can assist in a workable exchange that might further alert participants of particular users or unauthorized parties. This can prevent denial of service attacks and similar misuse of network traffic. Conversely, the exchange might maintain histories of the effectiveness of particular routes or particular parties that command a premium price or similar consideration for its recognition or reputation. For these reasons, an open form of rights and responsibility management—as opposed to traditional notions of access restriction-based digital rights management, are enabled for data—aesthetic, or functional content owned by other parties or for which rights need to be cleared.

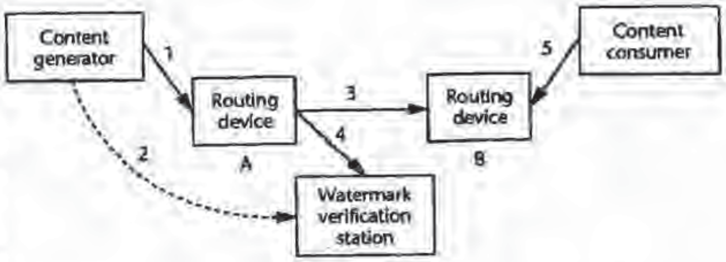


*Figure 1. System schematic.*

### Bandwidth provisioning

The packet watermark can help classify a data stream for a particular QoS. The data stream might be organized into a number of packets, and the sender can add a watermark to each packet's header. The size of the watermark can vary, but for illustration, a 32-bit watermark is stored, in say, the stream ID option field (that is, in the header) in the IPv4 packets. Preferably, the same 32-bit watermark would be placed in each and every packet in the flow. Additionally, for this example, the watermark's four most significant bits (MSBs) could help identify the QoS level, yielding 16 available levels, and the remaining 28 bits of the watermark could then uniquely identify the flow. One possible implementation for the remaining 28 bits is to store a unique identifier associated with a watermark packet key.

For example, the sender could create an array of the flow's secure hashes (for instance, SHA-1 or any hashing protocol deemed secure by the party or parties) using the watermark packet key. The watermark packet key, the watermark, and a portion of the flow make up the input to a hash function. The flow associated with one, two, or even more data packets could make up the portion of the flow used as input to the hash function. For this discussion, I consider the flow associated with one packet (that is, the portion of the flow inserted into one TCP/IPv4 packet). The hash's output might have a predetermined number of bytes. The array is the set of all hash outputs generated using successive portions of the flow until the complete flow has been processed. The outputs of the hash, the watermark packet key, and watermark are combined to create the watermark identification (WID).

Accordingly, the watermark can be matched to a corresponding WID. The component parts of the WID then help check the flow's authenticity. Moreover, if a portion of the watermark helps identify a particular QoS level, then we can evaluate the data for compliance for a particular path (such as for transmission by a compliant router). For higher security requirements, we can easily implement additional security protocols or tiered verification. This example uses four MSBs to identify a QoS level. This is simply a suggested format. Any predetermined bits can be used. It's preferable, however, that the same watermark be used within each packet of the stream. The watermark might not contain a QoS indicator, in which case, all bits allocated for the watermark might be used for a unique identifier, such as that associated with a particular watermark packet key. Figure 1 shows a schematic of how the system routes packets.

The WID holds all the dependent data. There's only one 32-bit watermark assigned for each stream and one WID created. The watermark packet key may be reused. So the WID might contain a

- 32-bit watermark, inclusive of any QoS indicator,
- watermark packet key,
- hash output from the first block of the flow of data steam,
- hash output from the second block of the flow,
- hash output from the third block of the flow, and
- a series that is bounded by the last block (the flow has a variable length depending on what the data represents).
- hash output from the last block of the flow.

Each router along the flow's path can read the watermark and determine its QoS by using those bits associated with the QoS indicator. Each router can then take appropriate action for prioritizing or deprioritizing each packet. These actions might include choosing a path based on load, reliability, or latency or buffering lower priority packets for later delivery.

The router configuration might enable checking each packet's authenticity. Preferably, the router configuration indicates checking a subset of the packets for authenticity and scaling up to additional cryptographic protocols thereby maintaining overhead or reducing computational requirements by adjusting security policy consistent with the authentic packet flow. For example, copies of a predetermined, small percentage of watermarked packets might be diverted to a sniffer. Preferably, the sniffer has received the WIDs for all authorized flows either before receiving the flows or in the same time frame. The sniffer compares the watermark of the copied packet to its WID table to find the appropriate WID. If the sniffer doesn't identify a corresponding watermarking key, it deems the packets unauthorized and instructs the router to deprioritize or, preferably, block the flow of the nonauthentic data. If the sniffer finds a corresponding WID, it calculates a hash output for the packet and attempts to match it to the corresponding hash in the WID. If the hash values match, the sniffer instructs the router to permit the flow to continue on its path. If the hash values don't match, the sniffer deems the packets nonauthenticated and notifies the router. Further rules might be associated with any number of scenarios as to why the router has deemed the flow nonauthentic, including notification and reference of the action to a database.

Ideally, the watermark generator software maintains a specific list of sniffers to receive the WID. For each of these, the WID should be sent encrypted and signed, using a public key technology such as PKIX certificates or Open PGP keys. One possible arrangement is having the watermark generator deliver the WID to trading partners who have established a prior business arrangement. The trading partners would pass the WID along to additional devices, eliminating scaling problems on the sender side. These might comprise, moreover, functions handled by the exchange and clearinghouse features.

Generally, it's advantageous for a sniffer to collect twice the original number of bytes to guarantee enough data to calculate a hash, given that the sniffer doesn't know *a priori* the original number of bytes. For large flows, 100:1 ratios might create unacceptably large WIDs. However, as the ratio decreases, the WID delivery channel gets larger. As the ratio increases, the amount of original content necessary to the sniffer increases, as does the amount of the flow that can pass before completion of an authorization check. Making the ratio sensitive to data type and size, or some predetermined policy parameters, dynamically optimizes the system to meet the needs of a particular market. Given this flexibility, overhead will more than likely remain small, compared to more granular accounting and its associated cost savings. Essentially, decisions concerning how much security should be mapped to the flow (for instance, applying a digital signature instead of a hash) are likely to mirror the business models of the markets for which packet watermarking is directed. To more fully extend the benefits of this example, later work will consider additional novel features concerning data management, pricing mechanisms, clearinghouse and dispute resolution methods, and systems.

## Conclusions

For electronic networks, any number of data files can occupy bandwidth at some discrete instance of time. The purpose of packet watermarking is twofold. First, it lets bandwidth-control devices recognize traffic that should move through the public Internet on specific paths, with either higher-than-normal or lower-than-normal priority. Second, the watermarking lets a bandwidth delivery service monitor its traffic to identify specific content sources. This is for purposes of revenue generation, content or data license management, bandwidth as payment or currency, or any other application where a specific data source needs identification. Watermark sampling requires two pieces of information, or the WID: the watermark key and the labeling information that associates the specific content with a hash array. The distribution of this information requires a secure mechanism because it contains cryptographic material (that is, the watermark key).

Security, like insurance, is a process for managing risk. Cryptographically identifying users demanding packets and subsequently provisioning a particular authenticated path (flow) between users is a basis for enabling bandwidth as currency. Heuristics might be applied as the system learns the best paths for packets to effectively determine subsequent use. Taken to another level, the packets can be further analyzed based on the data's nature, if such identification is available. Packet watermarks and data object watermarks establish responsibility for data's objects or functions (for algorithmic data, such as source, object and executable code). Such responsibility and accountability lie at the heart of a commercially acceptable platform for information commerce. **MM**

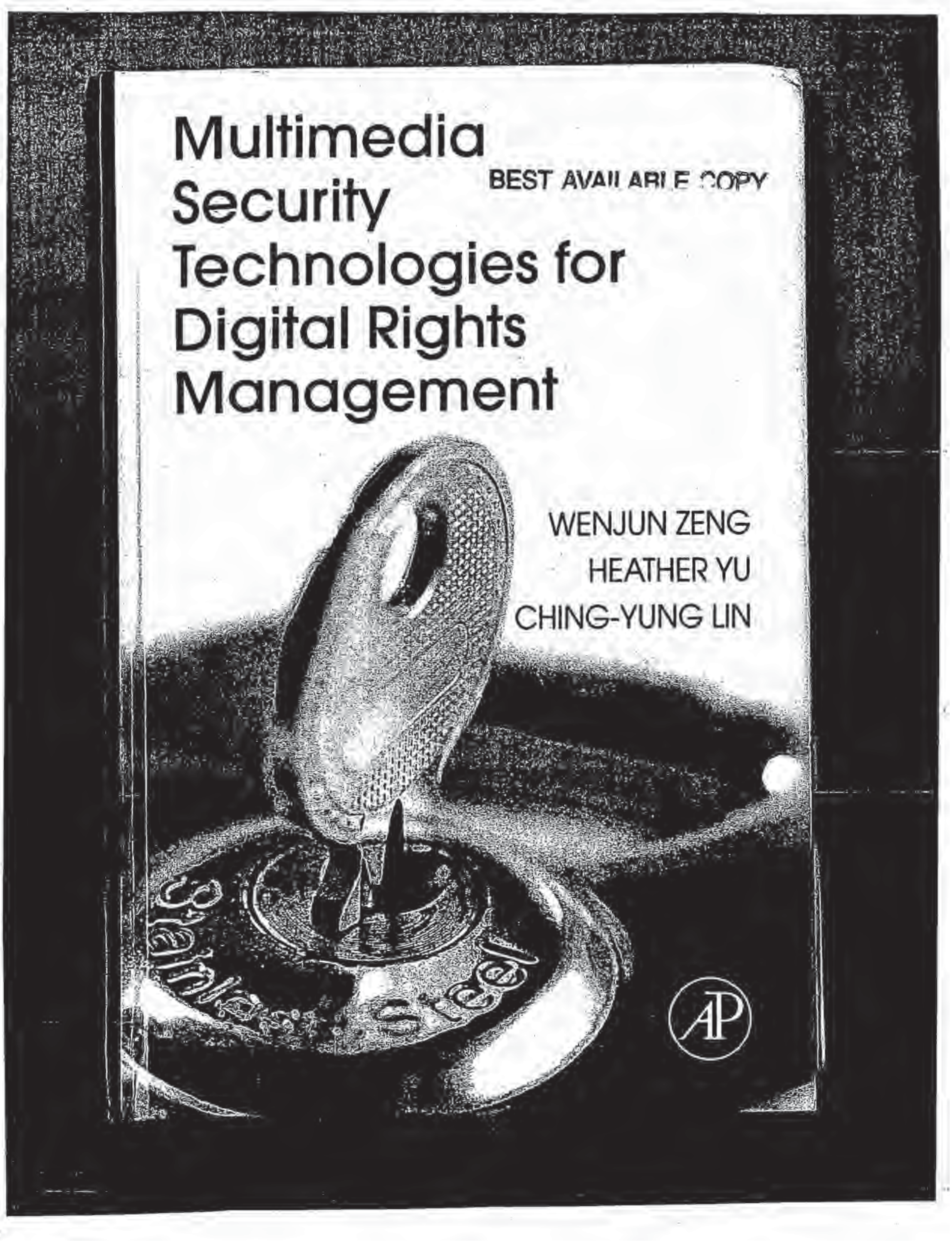
## Acknowledgments

Thanks to Mike Berry, Peter Cassidy, Nevenka Dimitrova, Yair Frankel, and Rodney Thayer for their helpful contributions to this work.

## References

1. G. Simmel, *The Philosophy of Money*, D. Frisby ed., originally published 1907, Routledge, 1999.
2. L. Weschler, *Boggs: A Comedy of Values*, The Univ. of Chicago Press, 1999.
3. A. Danto, *Transfiguration of the Commonplace*, Harvard Univ. Press, 1996.
4. S. Moskowitz, *So This is Convergence?*, Serendip, 1999 (in Japanese), http://www.bluespike.com/papers/convergence.pdf.
5. A. J. Menezes et al., *Handbook of Applied Cryptography*, CRC Press, 1997.
6. *Exchange Mechanisms for Digital Information Packages with Bandwidth Securitization, Multichannel Digital Watermarks, and Key Management*, US Patent Application Serial No. 08/674,726, Patent and Trademark Office, 1996.
7. *Optimization Methods for the Insertion, Protection, and Detection of Digital Watermarks in Digitized Data*, US Patent No. 5,889,868, US Patent and Trademark Office, 1999.
8. *Z-Transform Implementation of Digital Watermarks*, US Patent No. 6,078,664, US Patent and Trademark Office, 2000.
9. L. Lessig, *The Future of Ideas*, Random House, 2001.
10. S. Leibowitz, *Policing Pirates in the Networked Age*, Policy Analysis No. 438, Cato Institute, 2002.
11. J. V. DeLong, "Defending Intellectual Property", *Copy Fights: The Future of Intellectual Property in the Information Age*, Cato Institute, 2002.
12. *Method for Combining Transfer Functions with Predetermined Key Creation*, US Patent Application Serial No. 09/046,627, 1998 (notice of allowability issued).
13. *System and Method For Permitting Open Access To Data Objects And For Securing Data Within The Data Objects*, US Patent Application Serial No. 09/731,039, Patent and Trademark Office, 2000.
14. *Steganographic Method and Device*, US Patent No. 5,613,004, Patent and Trademark Office, 1997.
15. *Method for Stego-Cipher Protection of Computer Code*, US Patent No. 5,745,569, Patent and Trademark Office, 1998.
16. J. Bughin, "Black-Scholes Meets Seinfeld", *The McKinsey Quarterly*, no. 2, 2000, pp. 13-16, http://www.mckinseyquarterly.com/article_page.asp?ar=810&L2=17&L3=66.

*Readers may contact Scott Moskowitz at Blue Spike, 16711 Collins Avenue, No. 2505, Miami Beach, Fla. 33160, email scott@bluespike.com.*



# Multimedia Security Technologies for Digital Rights Management

WENJUN ZENG
HEATHER YU
CHING-YUNG LIN

Academic Press is an imprint of Elsevier
30 Corporate Drive, Suite 400, Burlington, MA 01803, USA
525 B Street, Suite 1900, San Diego, California 92101-4495, USA
84 Theobald's Road, London WC1X 8RR, UK

This book is printed on acid-free paper. ♾

**Library of Congress Cataloging-in-Publication Data**
Multimedia security technologies for digital rights management/edited by Wenjun Zeng, Heather Yu, and Ching-Yung Lin.
p. cm.
Includes bibliographical references and index.
ISBN-13: 978-0-12-369476-8 (casebound : alk. paper)
ISBN-10: 0-12-369476-0 (casebound : alk. paper) 1. Computer security. 2. Multimedia systems–Security measures. 3. Intellectual property. I. Zeng, Wenjun, 1967- II. Yu, Hong Heather, 1967- III. Lin, Ching-Yung.

QA76.9.A25M875 2006
005.8–dc22
2006003179

**British Library Cataloguing-in-Publication Data**
A catalogue record for this book is available from the British Library.

ISBN 13: 978-0-12-369476-8
ISBN 10: 0-12-369476-0

For information on all Academic Press publications visit our Web site at www.books.elsevier.com

Printed in the United States of America
06 07 08 09 10 9 8 7 6 5 4 3 2 1

# Table of C

# 1

# Introduction—Digital Rights Management

**Scott Moskowitz**

## 1.1 PROPERTY AND VALUE

Real property is familiar to most people. We live in houses, work in offices, shop at retailers, and enjoy ball games at stadiums. In contrast with "personality," which includes personal effects and intellectual property, real estate derives from *realty*—historically, land and all things permanently attached. Rights, whether for real property or intellectual property, have communal roots. Security, however, is a term with very subjective meaning. Simply "feeling secure" is not necessarily equivalent with the expectations or actual protections provided. Securing real property can mean locking a door or, for the significantly more paranoid, deploying tanks on one's lawn. Although it can be argued that intellectual property is related to real property, there are inherent and significant differences—the obvious one being that intellectual property is not physical property. The most controversial aspect of intellectual property is the ease at which it can be and is shared. Divergent viewpoints on this issue exist. At the extremes, "information is free," while others assert theft. We will leave the ability to define "piracy" to economists, lobbyists, policymakers, and even jurists with such interests. Clearly, we need to consider the law and the cost of copy protection when making technical decisions about designing the appropriate system. A particular set of problems will need definitions in order for agreement on any "secure" solutions. For this reason, any resource on "Digital Rights Management" (DRM) should include appropriate context. While other chapters of this book focus on technology topics and the development of the burgeoning market for DRM products and services, this chapter covers a number of topics identifying the importance of rights management technologies.

## 1.2 "ORIGINAL WORK"

It is prudent to provide a cursory outline of copyrights, not in the interests of providing any form of legal advice, but to delineate the impact of how copyright protection has evolved with respect to U.S. copyright law.[1] Copyright is established in the U.S. Constitution. The single occurrence of the word "right" in the Constitution appears in Article 1, Section 8, Clause 8: "[t]o promote the Progress of Science and useful Arts, by securing for limited times to authors and inventors the exclusive *right* to their respective writings and discoveries." As with all U.S. laws, the U.S. Congress first enacts legislation, while the courts provide judicial oversight and interpretation of law. Over time, legislation has been adopted making copyright more consistent with advances in the technology landscape. Lobbying efforts by a variety of stakeholders have provided additional impetus for change for economic reasons. Litigating "copyright infringements" represent additional efforts at defining copyright and its associated protections. However, when one has a copyright, what exactly does that mean? Essentially, a copyright is a form of contract between the creator of the original work and the public. While based on the recognition of property rights, in general, the creator agrees to make his work publicly available in consideration of legal recognition under the law. The Constitution promulgated copyright in the interests of promoting science and the arts for the benefit of society. Subsequent changes, challenges, and context have become arguably more public with the huge success of the Internet and networking technologies in general.

To be a bit more specific, a "work," the copyrighted value to be protected, is "created" when it is fixed in a copy or phonorecord for the first time: where a work has been prepared over a period of time, the portion of it that has been fixed at any particular time constitutes the work as of that time, and where the work has been prepared in different versions, each version constitutes a separate work. A "derivative work" is a work based upon one or more pre-existing works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or any other form in which a work may be recast, transformed, or adapted. A work consisting of editorial revisions, annotations, elaborations, or other modifications which, as a whole, represent an original work of authorship is a derivative work. As electronics and digital editing software become the inexpensive tools of the

---

[1] For international copyright issues, one helpful resource is http://caselaw.lp.findlaw.com/data/constitution/article01/39.html.

---

Information Age, co
argue the merits of s
we got here from the

## 1.3 LOOKING B

Including a list of t
National Informatio
tions to the Copyrig
networks such as th
timeline from which
the companies liste
purposes, it is not
it is helpful to prov
of technology impa
with legal rights. W
is referred to as "F
standard for establi
somewhat emblem
property.

> In Feist [Feist Pub
> the court explaine
> The primary obje
> mote the Progress
> the right to their
> ideas and informa
> Feist, 499 U.S.
> assures authors t
> is necessarily co
> protection, we do
> original expressi
> protection by 10

Section 107 of
the wide range of
works. Perhaps in
ing "security" inc
layered security
copyright and its
"fair use." Boun
the Copyright A
presents the mos

Information Age, copyright is thought to need additional protections. We do not argue the merits of such a belief, but provide the following milestones as to how we got here from there.

## 1.3 LOOKING BACK AT THE COPYRIGHT ACT OF 1976

Including a list of burgeoning "copyright protection" software companies, the National Information Infrastructure Copyright Act of 1995 made recommendations to the Copyright Act of 1976 and addressed the potential problems with open networks such as the "Internet." It is a fairly interesting point to start a historical timeline from which rights management technologies have evolved as several of the companies listed in that report made subsequent impacts in the field. For our purposes, it is not necessary to interpret the large body of legal arguments, but it is helpful to provide what limits have been argued and how far the perception of technology impacts DRM. After all, the copyright holder is not the only party with legal rights. While copyright previously concerned "sweat of the brow," what is referred to as "Feist," a modicum of creativity has become the more stringent standard for establishing copyright. An early case, *Lotus Corporation v. Borland* is somewhat emblematic of the early fights over copyright protection of intellectual property.

> In Feist [Feist Publications, Inc. v. Rural Telephone Serv. Co., 499 U.S. 340 (1991)], the court explained:
>
> The primary objective of copyright is not to reward the labor of authors, but to promote the Progress of Science and useful Arts. To this end, copyright assures authors the right to their original expression, but encourages others to build freely upon the ideas and information conveyed by a work.
>
> Feist, 499 U.S. at 349-50. We do not think that the court's statement that "copyright assures authors the right to their original expression" indicates that all expression is necessarily copyrightable. While original expression is necessary for copyright protection, we do not think that it is alone sufficient. Courts must still inquire whether original expression falls within one of the categories foreclosed from copyright protection by 102(b) [1].

Section 107 of the Copyright Act of 1976 provides additional guidance for the wide range of stakeholders who may need to access or manipulate copyrighted works. Perhaps inevitably, reverse engineering and related attempts at circumventing "security" increase the perception that copies of the original work may require layered security and additional legal protections. The least understood aspect of copyright and its place "to promote the Progress of Science and useful Arts" regards "fair use." Bounded by several factors, the relative weights are not provided by the Copyright Act of 1976, and fair use may indeed be the one legal issue that presents the most difficult challenges in engineering solutions to piracy.

> Four factors must be considered: (1) the purpose and character of the use, including whether such use is of a commercial nature or is for non-profit educational purposes; (2) the nature of the work; (3) the amount and the substantiality of the portion used in relation to the copyrighted work as a whole; and (4) the effect of the use on the market value of the copied work [2].

The one case at the heart of the most extreme debates in copyright circles may be *Sony Corporation v. Universal City Studios* (1984), concerning the sale of videocassette recorders (VCRs). The U.S. Supreme Court ruled that "[b]ecause recorders were 'widely used for legitimate, unobjectionable purposes,' the recording did not constitute direct infringement of the studio's copyrights .... Absent such direct infringement, there could be no contributory infringement by *Sony* [3]." The key factor being that there was value in personal recording. While citing the concept of fair use, which protects consumers from *some forms* of copyright infringement, the debate did not end with this ruling. Indeed, the concept of fair use has been extended to areas not previously anticipated, including reverse engineering of copyrighted software.

Additionally, the Copyright Act of 1976 laid several other "foundations," though they are still unsettled in the minds of the stakeholders involved. Besides extending the length of copyright protection, library photocopying was changed to make possible preservation and inter-library loans without permission. Section 107 is at the heart of the types of issues for evaluation of DRM system design, even if less than all stakeholders' rights are considered. Fair use is a doctrine that permits courts to avoid rigid application of the copyright statute when to do otherwise would stifle the very creativity that copyright law is designed to foster. One author addresses this notion of relativity in the early days of the Internet Age.

> The doctrine of fair use recognizes that the exclusive rights inherent in a copyright are not absolute, and that non-holders of the copyright are entitled to make use of a copyrighted work that technically would otherwise infringe upon one or more of the exclusive rights. Although fair use originated 'for purposes such as criticism, comment, news reporting, teaching, ... scholarship, or research,' it also applies in other areas, as some of the examples below illustrate. However, courts seem more willing to accept an assertion of fair use when the use falls into one of the above categories. Perhaps more than any other area of copyright, fair use is a highly fact-specific determination. Copyright Office document FL102 puts it this way: 'The distinction between "fair use" and infringement may be unclear and not easily defined. There is no specific number of words, lines, or notes that may safely be taken without permission. Acknowledging the source of the copyrighted material does not substitute for obtaining permission.' The document then quotes from the 1961 Report of the Register of Copyrights on the General Revision of the U.S. Copyright Law, providing the following examples of activities that courts have held to be fair use:—Quotation of excerpts in a review or criticism for purposes of illustration or

comment;—Quotati
tration or clarificati
content of the work
in a news report;—
a damaged copy;—
illustrate a lesson;—
reports;—Incidental
located in the scene

Several other mor
provide a broader co

**Digital Millennium**
is the provision,–
tion of access res
the copyright ow
in place to prote
However, it is st
sures can be circ
computer, as in th
action that is inh
ian of Congress
Congress.
**Digital Theft Deter**
Congress increas
from that of $500
from $100,000 to
**Librarian of Cong**
of Congress issu
Circumvention P
two exemptions i
by filtering soft
programs and da
permit access be
ommendation can
**Dmitri Skylyarov**
grammer for Ele
Reader DRM. Al
tinued with the p
the DMCA. As o
observers viewe
could be pushed
"not guilty" in la

comment;—Quotation of short passages in a scholarly or technical work for illustration or clarification of the author's observations;—Use in a parody of some of the content of the work parodied;—Summary of an address or article with brief quotations, in a news report;—Reproduction by a library of a portion of a work to replace part of a damaged copy;—Reproduction by a teacher or student of a small part of a work to illustrate a lesson;—Reproduction of a work in legislative or judicial proceedings or reports;—Incidental and fortuitous reproduction in a newsreel or broadcast, of a work located in the scene of an event being reported [4].

Several other more recent legal and legislative actions should be mentioned to provide a broader consideration of what the fuss is really all about.

**Digital Millennium Copyright Act, the "DMCA" (1998).** Key among its impact is the provision, known as Section 1201, of a prohibition on circumvention of access restriction controls or technological protections put in place by the copyright owner. If a copyright owner puts an access restriction scheme in place to protect a copyright, unauthorized access is essentially illegal. However, it is still unclear how to define "access restriction" if such measures can be circumvented by holding the shift key at start-up of a personal computer, as in the case of one access restriction workaround or any consumer action that is inherent to the use of general computing devices. The Librarian of Congress conducted a proceeding in late 2000 to provide guidance to Congress.

**Digital Theft Deterrence and Copyright Damages Improvement Act (1999).** Congress increased damages that can be assessed on copyright infringements from that of $500 to $750 to $20,000 to $30,000. Willful infringement increased from $100,000 to $150,000.

**Librarian of Congress Issues Exemptions to the DMCA (2000).** Librarian of Congress issues exemptions to the DMCA, Section 1201(a)(1), the Anti-Circumvention Provision, for "classes of works" that adhere to fair use. These two exemptions include: "Compilations consisting of lists of websites blocked by filtering software applications; and Literary works, including computer programs and databases, protected by access control mechanisms that fail to permit access because of malfunction, damage, or obsoleteness." The full recommendation can be found at http://www.loc.gov/copyright/1201/anticirc.html.

**Dmitri Skylyarov Arrested under DMCA Provisions (2001).** The Russian programmer for ElcomSoft was accused of circumventing Adobe Systems' eBook Reader DRM. Although Adobe later reversed course, government attorneys continued with the prosecution of the case, presumably to test the interpretation of the DMCA. As one of the first criminal cases brought under the DMCA, many observers viewed this as a test case for how far allegations under the DMCA could be pushed into actual indictments. A federal jury returned a verdict of "not guilty" in late 2002.

**U.S. Supreme Court Hears Challenge to Sonny Bono Copyright Term Extension Act, the "CTEA" (2002).** In copyright debates Lawrence Lessig, a well-known constitutional scholar, has been active in promulgating such mechanisms as the "Creative Commons." His representation of the plaintiffs in *Eric Eldred v. John Ashcroft* extended his experience in the copyright debate. Ultimately, the Supreme Court ruled against the plaintiffs, affirming the constitutionality of the CTEA and affirming Congress's role in intellectual property. Retrospectively, the CTEA extended existing copyrights by 20 years—to 70 years from the life of an author, from 50 years. As well, adding 20 years of protection to future works. Protection was extended from 75 to 95 years for "works made for hire," a common contractual framework used by many corporations.

***MGM v. Grokster* (2005).** It is unclear how many rounds of dispute resolution between technology innovators and content owners will go before the courts or Congress. For this reason, it may take some time to understand fully the impact of the *MGM v. Grokster* decision. The most widely quoted aspect of the ruling, thus far, concerns who should determine when a device is "promoted" to infringe copyright. The Supreme Court essentially decided:

> For the same reasons that *Sony* took the staple-article doctrine of patent law as a model for its copyright safe-harbor rule, the inducement rule, too, is a sensible one for copyright. We adopt it here, holding that one who distributes a device with the object of promoting its use to infringe copyright, as shown by clear expression or other affirmative steps taken to foster infringement, is liable for the resulting acts of infringement by third parties. We are, of course, mindful of the need to keep from trenching on regular commerce or discouraging the development of technologies with lawful and unlawful potential. Accordingly, just as *Sony* did not find intentional inducement despite the knowledge of the VCR manufacturer that its device could be used to infringe, 464 U.S., at 439, n. 19, mere knowledge of infringing potential or of actual infringing uses would not be enough here to subject a distributor to liability. Nor would ordinary acts incident to product distribution, such as offering customers technical support or product updates, support liability in themselves. The inducement rule, instead, premises liability on purposeful, culpable expression and conduct, and thus does nothing to compromise legitimate commerce or discourage innovation having a lawful promise [5].

In the world of physical media distribution, there are many channels available, both for broadcast and for physical carriers. Specialized retailers compete for consumer sales by differentiating their efforts from other more generalized retailers. Written content and imagery attracts consumers to publications such as magazines; and spoken content and music selection attracts consumers to radio. The number of possible combinations of content and editorial material provides for rich broadcast opportunities, which have the effect of attracting advertising

dollars to the broac
type schemes are n
to grow over time,
consumers has grow
obvious aim of adv

The argument th
is beginning to me
for said consumpti
Supply meets dem
lular phone), band
CDs, books, and D
needs consideratio
ability to measure "
contentious, the ar
can technical contr
implementation an
and services handl
value in securing
who should determ
should be provided
for free? What c
provider's property

## 1.4 COMMUNI

When considering
First, multimedia d
digital signal proc
economic and mo
for any copies m
Manufacturing has
profit margins, bu
interests of those r

### 1.4.1 Shannon'

Before delving in
munications and
points. World Wa
militaries, govern
time of great techr

dollars to the broadcasters. The parallels with online streaming or pay-per-click-type schemes are not a coincidence. Total spending on advertising has continued to grow over time, although the ability to reach a profitable, aggregated group of consumers has grown more difficult. The ability to reach paying audiences is the obvious aim of advertising.

The argument that there is too much entertainment vying for consumers' dollars is beginning to meet the more complicated issue of how to measure actual time for said consumption, while deploying efforts at protecting copyrighted material. Supply meets demand whether measured in units of time (e.g., minutes on a cellular phone), bandwidth (e.g., amount of data per unit of time), or copyrighted CDs, books, and DVDs. Some agreement on the unit of measurement obviously needs consideration. When supply is controlled, as with generalized DRM, the ability to measure demand may become distorted. Though the conclusions are contentious, the arguments can be made from a variety of viewpoints. Simply, can technical controls for accessing copyrighted material cost less than the cost of implementation and maintenance of these same controls? How are new devices and services handled given legacy control systems or even open systems? Is there value in securing copyrights with DRM? What rights of revocation exist, and who should determine the scope and form of revocation? How much open access should be provided to consumers? Is there value in providing copyrighted works for free? What constitutes a consumer's property in contrast with a content provider's property?

## 1.4 COMMUNICATION THEORY—WHO SCREAMS LOUDEST?

When considering the security of multimedia data, several issues pose challenges. First, multimedia data is compressible and easily transferable. Second, advances in digital signal processing have made the ability to digitize analog waveforms both economic and more commercially viable. Third, ownership and responsibility for any copies made of digitized content are typically a double-edged sword. Manufacturing has been made inexpensive to the owners and licensors, increasing profit margins, but content has increasingly been copied without regard to the interests of those rights holders. More on these issues will be discussed below.

### 1.4.1 Shannon's Gift

Before delving into technical aspects of DRM, attention must be paid to communications and cryptography. Cryptography has impacted history at several points. World War II was emblematic of the tight relationship between codes, militaries, governments, and politics—before the first microprocessors, but at a time of great technical innovation. The work in cracking the codes of that war was supplemented later by a growing interest in the underlying nature of communications. Largely unknown to the public, the seminal work of Claude E. Shannon in *The Mathematical Theory of Communication* and *Communication Theory of Secrecy Systems* provides helpful analysis in what can be expected theoretically. Developments based on communication theory, including cryptographic systems, are pervasive in modern society. The impact on our daily lives is incalculable. Telephones, financial markets, and even privacy itself have changed in dramatic, often unpredictable, ways. The demand for codes to assist with the secure transport of sensitive data was matched by the increasing importance of computerized networks for dispersal and distribution of such data.

At some point, confidentiality, one of several primitives designed into data security systems, was met by increasing calls for restrictions on the deployment of cryptographic protocols. Separately, but just as important, authentication, data integrity, and non-repudiation—additional primitives of cryptography—assisted in the growth of business over electronic networks. Public key cryptography provides all four of these primitives, in a manner making distribution of codes and ciphers economically feasible for all persons wishing to secure their communications. The landmark failure of the U.S. government's Clipper chip [6] in 1993 was only the beginning of an increased public interest in cryptography. With the proliferation of more bandwidth and anonymity, in many cases based on so-called strong encryption, commercial concerns were also heightened. Here, we deal specifically with copyrighted works such as images, audio, video, and multimedia in general. A basic notion that should be considered in understanding DRM may well be how to balance privacy with notions of piracy. Ironically, the emphasis on protecting privacy has been trumped in many ways by the goal of securing against piracy. Should personal secrets be shared to satisfy the demands of copyright holders? Put another way, is a social security number used to secure a purchase for a song download a fair exchange of value asserted by the copyright holder?

Shannon's conceptualization of communication theory provides a fitting background to copy protection techniques to be explored in this book. Actual performance of real-world systems should be matched against theory to encourage appropriate expectations. Communication theory at its most basic level is about the transmission of information between a sender and a receiver. The information typically has meaning or context. Obviously, there are limitations to communication systems as explored by Shannon and others. The channel and destination of the information being transmitted provide additional parameters to a communication system. Here, we eliminate the simplified arrangements for a noiseless communication channel where the inputs and outputs are equivalent. By noiseless we mean no "chance variables" occur, and thus no redundancy or other error correction is needed to communicate messages.

The ratio of the actual rate of information transmission to capacity in a given channel is called the efficiency of the coding scheme. Efficiency to both the sender

and the receiver can hav
istic scheme is analyzed,
it is proven that there ar
sets of errors (which car
Because binary data is e
each bit of data in the a
with 1 or 0 being the lim
elements of the coin flip c
munication, the entropy
source, the entropy of th
tion, correction, and conc
with the context of the i
and the entropy of the
may be successfully rep
part, Shannon's legacy,
cern ourselves with how
approximate the original
processing and in a philc
replica, but is an exact
mated waveform. The n
value of the coding sche
economics of deployme

In a discrete channel
a "chance variable," w
precise digitization sys
communicate informati
hardware systems. As
senders and receivers ca
of functions that facilita
ted. Similarly, the ense
the presence of noise i
observers. So long as th
associated message ca
by a cryptographic alg
tationally easy to disc
cipher.

The key is thus a st
receiver can be assured
sage. The data transm
communication chann
tion error (e.g., what is
be small relative to th

and the receiver can have subjective measurements as well. When a more realistic scheme is analyzed, namely efficient transmission in the presence of noise, it is proven that there are still a finite number of errors (perceptibly "noise") or sets of errors (which can be mathematically generalized to create noise filters). Because binary data is either a "1" or a "0" in a given channel, we can say that each bit of data in the abstract may be completely random by flipping a coin, with 1 or 0 being the limited choices. That is not to say that entropy of any of the elements of the coin flip can be ignored. However, in order to ensure effective communication, the entropy of any chance variables, the entropy of the information source, the entropy of the channel, etc. must be taken into account. Error detection, correction, and concealment form a large body of work in dealing specifically with the context of the information, the channel and nature of the transmission, and the entropy of the source impacts the channel capacity. That information may be successfully reproduced and can be expressed mathematically is, in large part, Shannon's legacy. This applies to cell phones and DVDs. Here, we concern ourselves with how a perceptible signal can be digitized, or "sampled," to approximate the original analog waveform. However, as is well known in signal processing and in a philosophical sense, the digitized signal can never be a perfect replica, but is an exact facsimile of an otherwise analog and infinitely approximated waveform. The natural limit is quantization itself; however, the limit of the value of the coding scheme in terms of practical use is human perception and the economics of deployment.

In a discrete channel, entropy measures in an exact way the randomness of a "chance variable," which itself may be random. The development of very precise digitization systems representing an "ensemble of functions" used to communicate information has been reduced into a multitude of software or hardware systems. As we delve into cryptography, here, we quickly note that senders and receivers can exchange secrets, or "keys," associated with an ensemble of functions that facilitate agreement over the integrity of the data to be transmitted. Similarly, the ensemble of functions assures transmission of the message in the presence of noise in the channel. Keys may be mistaken as noise by other observers. So long as the sender and receiver can agree to the key, the "secret," the associated message can be authenticated. The key is ciphered (i.e., processed by a cryptographic algorithm) in a manner to mimic randomness not computationally easy to discover even if the other observers are in possession of the cipher.

The key is thus a state or index of an ensemble of functions from which the receiver can be assured that the sender of the message did indeed transmit the message. The data transmission's discrete rate may not exceed the capacity of the communication channel. Finally, relating back to sampled signals, the quantization error (e.g., what is related to data conversion between analog to digital) must be small relative to the information transmitted in order to establish sufficiently small probabilities that the received signal is the communication intended by the sender. Statistically isolating "perturbing noise" from other errors and bounding upper and lower limits of capacity in a communication channel are presently computationally easy.

The introduction of digital CDs resulted from agreements over trade-offs of the general technologies so far described. As a medium for music, it is fitting to observe this medium for rich discussions on DRM. The CD is itself a discrete communication channel. The reflective material sandwiched between transparent plastic, which can be read by a CD player, is converted into a series of binary data (1s and 0s) as physical pits on the reflective material substrate. This data stream has pre-determined sampling rates and quantization values (16 bits, 44.1 kHz per second, for a Red Book Specification Audio Compact Disc). Again, data bits which have pre-determined locations or modality on the physical CD, are fed through an ensemble of functions which filter the digitized sample information stream into analog audio signal data. This data, of course, may be compressed for more economic use of bandwidth. We hear a song, the binary information sent out to an amplifier to be transduced, but, there is no "perceptually obvious" relationship with the music rendered. The data are presented according to the Red Book standard. We hear the music with our psychoacoustic abilities, our ears, and ultimately, our brains process the music and may associate the music information with some other independent or unrelated information.

Any such "associated information" may be different for every listening experience, every time for every individual listener. We would call this associated information "value added" or "rich" because it can be associated, with other independent information that may have no relationship with the primary communicated information which is the same for all listeners. The "hits" are hits for each individual in different ways that are aggregated in such a manner that they can be called hits—the memorable song for a high school prom, the one played when waking up, or any number of events associated with the copyrighted work in unintended ways, impacting the value attributed to such a work. Money is one obvious measure of success. Acting out a song may reflect the meaning intended by its creator or it may not. What matters with regards to DRM are the decisions made by creators and consumers of copyrighted works to create, seek, and consume with a fixed and limited amount of time and money determined by the harsh realities of the marketplace. Recognizable and potentially valuable multimedia can be rendered by general computing devices. Multimedia having many different interpretations depending on what stake the party has in the work. After all, creators, too, may give their work away for free.

We have generalized that it is computationally feasible to reproduce information, allowing senders and receivers to share the gestalt of information that may be transmitted. We ignore the specifics of digital filters and error correction to stress the point that, conceptually, data can be communicated and

communicated sec
on bandwidth or c
book, the cost of
data. Additionally,
high, certain other
transmit over con
by extension, digit
original analog wa
bandwidth [7].

Interestingly er
mation transmissi
tems must be ec
authentic or genu
source is trusted
play a role in es
mation, when "e
communications
acceptable fidelit
(i.e., "RMS," to
quency weightec
components pric
data through a
put), absolute er
perception (whic
is received by ou
discrete case (di
input data).

### 1.4.2 Kerckh

In cryptography
in order to prov
systems themsel
by providing pa
making the mos
A "keyed" algo
of the ensembl
of all keys defi
ated by a speci
input, the use o
operates as a T

communicated securely. If the communication channel is too expensive, based on bandwidth or overall available transmission capacity or, as is central to this book, the cost of protection, it ceases to play a role in enabling security of data. Additionally, if the bandwidth requirements for reproduction are sufficiently high, certain other types of data are not computationally feasible to economically transmit over communication channels. As more information is digitized and, by extension, digitally copied, even if there are imperceptible differences with the original analog waveform, the limit to data transmission becomes closely linked to bandwidth [7].

Interestingly enough, Shannon does address "intelligibility criterion" of information transmissions in providing "fidelity evaluation functions." Because systems must be economically practical, and information is ultimately deemed authentic or genuine by the creator or source of the information (assuming the source is trusted or the information can be verified), human perception does play a role in establishing a close enough proximity of replicated data information, when "exact recovery" is infeasible, given the presence of noise in communications channels. The five examples Shannon provides for measuring acceptable fidelity of a proposed information channel include root mean square (i.e., "RMS," to assist in determining coordinate information of the data), frequency weighted root mean square (essentially weighting different frequency components prior to RMS, which is similar to passing the distance between data through a shaping filter and calculating the average power of data output), absolute error criterion (over the period of zero to a discrete time), human perception (which cannot be defined explicitly, though we can observe how noise is received by our senses and our brain, sufficiently subjective parameters), and the discrete case (differencing input from output and dividing by the total amount of input data).

### 1.4.2 Kerckhoffs' Limits

In cryptography, the content or bits comprising the message must not be changed in order to provide acceptable levels of confidence in a secure system. However, systems themselves cannot guarantee security. A human can compromise a system by providing passwords or systems may generate weak pseudo-random numbers, making the most seemingly strong "cryptographic algorithm" ("cipher") unsecure. A "keyed" algorithm defines an ensemble of functions with the specific member of the ensemble identified by a unique key. With respect to encryption, the set of all keys defines a plurality of encryption functions. Each element is instantiated by a specific key. Though there may be randomness ("entropy") within the input, the use of the randomness only relates to the manner in which the function operates as a Turing machine (e.g., a general computing device). The random choice of a key to specify the element in the plurality of encryption functions is essential.

As Shannon stressed, communications is concerned with "operations on ensembles of functions," not with "operations on particular functions." Cryptography, too, is about ensembles of functions. The basic difference with coding (i.e., communications) is the exchange of the key. The ensemble of functions occupies a finite set, so that the input and output can be secured by associating the data to be transmitted with a randomly generated key that is pre-determined by both parties by some mutually agreed to means—the cryptographic algorithm or cipher. Kerckhoffs' law is the foundation by which such determinations are made; it is assumed that the adversary possesses the cipher, and thus the security must rest in the key. Auguste Kerckhoffs provided five additional principles, including (1) system indecipherability, (2) the key must be changeable, (3) the system should be compatible with the means of communication, (4) portability and compactness of the system is essential, and (5) ease of use. Of these principles, ease of use and whether security rests with the key have historically made for difficult engineering challenges within DRM. In cases where DRM systems must come in contact with other DRM systems, these challenges are heightened. Some have argued that it is not possible to tamperproof cryptographic systems to sufficiently prevent hacks [8]. This has obvious impacts on DRM.

## 1.5 CRYPTOGRAPHY—MUCH TO DO

With a basic understanding of communications theory and its relationship with cryptography, we can describe two conventional techniques for providing key-based confidentiality and authentication currently in use: symmetric and asymmetric encryption. Both systems use non-secret algorithms to provide encryption and decryption and keys that are used by the algorithm. This is the basis for Kerckhoffs' law: all security should reside in the key, as it is assumed the adversary will have access to the cryptographic algorithm. In symmetric systems, such as AES, the decryption key is derivable from the encryption key without compromising the security of the message. To assure confidentiality and authenticity, the key should be known only to the sending and receiving entities and is traditionally provided to the systems by secure physical communication, such as human courier. Other systems where a common key may be developed by the sender and receiver using non-secure communications are widely deployed. In such systems, each party to a communication generates a numerical sequence, operates on the sequence, and transfers the result to the other party. By further operation using the transferred result and the locally generated sequence, each party can develop the identical encryption key, which cannot be obtained from the transferred results alone. As implemented for use over the Internet, common encryption systems are

---

those denoted by
protocols.
In asymmetric
a numerical sequ
ferent encrypting
to a second part
encrypting key (
message that ca
party. The key ge
not be derived fi
using non-recipr
digital signature
signature functic
manner parallel
key can be used
has been digitall
the originating p
origination of th
key. So, how dc
est to establish c
DRM system, m
dentiality. Howe
political constrai
or more generall
not possible to t
communications
practicality can
and "embedding

1.6 DIGITAL
AND EMI

It is not prudent
may not always
Rights are typic
stakeholders ma
extensions gene
related to encry
is evident in ana
previously, scra
the encryption
must be decrypt

those denoted by the Secure Socket Layer (SSL) and IP Security Protocol (IPSEC) protocols.

In asymmetric encryption systems, a first party to a communication generates a numerical sequence and uses that sequence to generate non-reciprocal and different encrypting and decrypting keys. The encrypting key is then transferred to a second party in a non-secure communication. The second party uses the encrypting key (called a public key because it is no longer secure) to encrypt a message that can only be decrypted by the decrypting key retained by the first party. The key generation algorithm is arranged such that the decrypting key cannot be derived from the public encrypting key. Similar methods are known for using non-reciprocal keys for authentication of a transmission. There are also digital signature algorithms. In some cases, as with RSA, encryption and digital signature functionality are properties incorporated by the same algorithm. In a manner parallel with the real-world handwritten signatures, the non-secure public key can be used to tamperproof a message (i.e., providing nonrepudiation) that has been digitally signed using a secure "private" or secret key known only to the originating party—the signer. Thus, the receiving party has assurance that the origination of the message is the party who has supplied the "public" decrypting key. So, how does this relate to DRM? We have devised several areas of interest to establish commonality of the elements typically considered in designing a DRM system, namely authentication, data integrity, non-repudiation, and confidentiality. However, DRM is inherently constrained from legal, economic, and political constraints, as well as consumer expectations—not strictly cryptography or more generally communication theory. Mentioned previously, some argue it is not possible to tamperproof software programs given the inherent foundations of communications. Within the DRM product and service space, terminology and practicality can vary widely. Here, we generalize DRM by discussing "wrapping" and "embedding," so-called "digital watermark," technology.

## 1.6 DIGITAL RIGHTS MANAGEMENT—WRAPPING AND EMBEDDING

It is not prudent to limit our discussion solely on word choice. Essentially, the terms may not always reflect the utility or functionality of the protections being described. Rights are typically matched by responsibilities. DRM offers up examples of how stakeholders may not share common interests [9]. Copy protection and content extensions generally apply to digitized content, while "scrambling," a scheme related to encryption, may be applied to an analog signal. Such analog scrambling is evident in analog cable and analog cell phone systems. Encryption, as discussed previously, scrambles content, but the number of 1s and 0s may be different after the encryption process. In some scenarios, prior to enabling access to content it must be decrypted, with the point being that once the content has been encrypted, it cannot be used until it is decrypted. Encrypted audio content itself might sound like incomprehensible screeching, while an encrypted image or video might appear as random noise when viewed. The encryption acts as a transmission security measure—access control. One approach has commonly been called "conditional access" when someone or something has the right to access the media. In many scenarios, identifying information or authentication of that party must first be completed prior to decryption of the content or description of the intended scope of use. There may be layered access restrictions within the same scheme. In either case, the transmission protection ends when the content is to be observed.

Encryption is poorly applied in at least two specific areas with respect to copy protection of content. First, so-called "pirates" have historically found ways to crack the protection as it is applied to content. The effect is essentially equivalent to obtaining the decryption key without paying for it. One such technique is "differencing," where an unencrypted version of the content is compared with an encrypted version of the same to discover the encryption key or other protections. Differencing is also a weakness in many digital watermark systems. In some watermark systems, the requirement to maintain original unwatermarked material for comparing and recovering embedded code from a suspect copy of content introduces other problematic issues such as additional data storage requirements at the detection side. Why store watermarked content for protection purposes when unwatermarked content may exist at the same site for decoding said watermarks? Second, and perhaps more complicated to address, is that once a single legitimate copy of content has been decrypted, a pirate is now free to make unlimited copies of the decrypted copy. In effect, in order to make, sell, or distribute an unlimited quantity of content, the pirates could simply buy one copy, which they are authorized to decrypt, and make as many copies as desired. These issues were historically referred to as the "digital copy problem"; others prefer "digital piracy."

Copy protection also includes various methods by which an engineer can write software in a clever manner to determine if it has been copied and, if so, to deactivate the software. The same engineer may be a "rogue engineer" who essentially has the backdoor key to deactivate the copy protection. This is typically the result of a poorly chosen encryption algorithm or means for obtaining a key. Also included are undocumented changes to the storage format of the content. Copy protection was generally abandoned by the software industry, since pirates were generally just as clever as the software engineers and figured out ways to modify their software and deactivate the protection. The cost of developing such protection was also not justified considering the level of piracy that occurred despite the copy protection. That being said, the expansion of software product activation keys, online registration schemes, and registered version upgrades indicates increased interest and benefit in securing even software programs. Software watermarking schemes, including those using "steganographic ciphers,"[1] have correspondingly increased over the past few years [10].

Content extension re
cating whether a copy
with regards to the use
system must be specifi
information and interp
system is the Serial C
Audio Tape (DAT) har
on the track immediate
it can be copied. The h
wrapping content, we
formalize concepts be

When we discuss
information in plain
They need not be mut
Watermarks [11] are
placing "transactional
transaction informatio
mation known by the
of the electronic copy
of the electronic copy
modification of the
ceptual quality of the
identifiable. More ad
with the system. Thi
content security syst
between the protectio
tent to be protected.
any wrapped, embed
or inaccessible. In pa
holders have yet to e
success solely throu

### 1.6.1 Who Is in C

Protection of copyri
tial of loss at the tim
after the fact to con
trols are complemen
Such consideration
is worth most for p
bution channels, an
greater reduction of

Content extension refers to any system attaching some extra information indicating whether a copy of the original content can be made or some other logic with regards to the use and accessibility of the content. A software or hardware system must be specifically built around this scheme to recognize the additional information and interpret it in an appropriate manner. An early example of such a system is the Serial Copyright Management System (SCMS) included in Digital Audio Tape (DAT) hardware. Under this system, additional information is stored on the track immediately preceding each sound recording indicating whether or not it can be copied. The hardware reads this information and uses it accordingly. By wrapping content, we are generally referring to "content extensions." We further formalize concepts below.

When we discuss watermarks, we are addressing steganography, or hiding information in plain view, in combination with cryptographic techniques. They need not be mutually exclusive and in many cases complement each other. Watermarks [11] are a unique technology that embed and protect a "code" by placing "transactional information" intrinsically within the electronic work. The transaction information can specify time, date, recipient, and supplementary information known by the transmitter at the time of the transfer to the recipient. Review of the electronic copy of the media at a later instance reveals the historical record of the electronic copy. Safeguarding from manipulation or deletion, unauthorized modification of the transactional information results in degradation of the perceptual quality of the work. Tampering with watermarked media is, thus, quickly identifiable. More advanced schemes include watermark code which itself interacts with the system. This code, with or without interaction with a key, can upgrade content security systems and can be characterized by a variety of interactions between the protection scheme, associated keys, watermark information, and content to be protected. Before delving into finer detail, we note that it is unclear that any wrapped, embedded, or generally "DRM'd" content has remained wrapped or inaccessible. In parallel, we have not observed clear examples where copyright holders have yet to eschew traditional distribution channels to achieve economic success solely through DRM distribution schemes.

### 1.6.1 Who Is in Control—Active and Reactive Controls

Protection of copyrighted works may be a proactive control that reduces the potential of loss at the time of an event, while a reactive control provides an audit trail after the fact to conclude what happened and by whom. The two types of controls are complementary and, in many cases, can and should be used concurrently. Such consideration as the time value of the content, that period in which the content is worth most for protection, is subjective and varies among media types, distribution channels, and market forces. Yesterday's newspaper arguably suffers far greater reduction of economic value than long-running hits on Broadway during the time it takes a new edition of the newspaper to appear (changes in critiques of the Broadway work, notwithstanding). Uniqueness over data or data copies assists in establishing responsibility for the data. Similar to the physical world use of receipts for transactions over the "same" material, watermarks act as a control for receipts of digitized data. However, time also plays a significant role in value.

Active controls provide a first line of defense in times of a breach in security. With regard to data security risks, there are several types of commonly established information security controls, generally categorized as physical, procedural, and logical controls. Physical controls are generally building access and alarm systems. Procedural controls include policies, operating procedures, training, and audits. Logical controls are placed at the computer system level and include application and operating system-level access controls, lists, and perimeter protection with firewalls, router security, and intrusion detection systems. With respect to the copying of copyrighted media, the most common type of active controls is "security wrappers," often called ("active") DRM [12]. A wrapper wraps the digital media around a digital structure to prevent extraction of the media from the stored data object. Generally, the wrapper includes encryption of the media, "meta-data" about the media, and may include other logic, encrypted or not. A simplistic explanation follows here.

First, content is encoded with associated meta-data, followed by encryption of the meta-data and media, and any additional non-encrypted data may be placed. Finally, additional information, oftentimes a software wrapper, that must be run to extract the media is added. The data object is stored directly within the software wrapper. That is, the media is blanketed with multiple layers of controls. To obtain the media in a perceptually similar form, the wrappers must be removed. Hence, this is an active control. However, to be useful, the wrapper must be removed, making the media extremely vulnerable at the time of use (viewing, playback, etc., when the media is "in the clear" and susceptible to unauthorized use). The software wrapper may also require active coordination by a third party during the unwrapping process. For instance, the software wrapper may require interaction with the content provider to obtain keys to decrypt the content. This communication requirement adds additional complexity to the process and, if required, places additional constraints when the active DRM-protected media is part of a larger workflow. Watermarks need not be incorporated in the previous example. Instead, meta-data are placed external to the content for operational requirements, and both the meta-data and the media are encrypted. The meta-data, for instance, may provide cryptographic authentication of the media or may provide keys for an external cryptographic operation that must be performed again including upgrades to the system in parts or in its entirety. The placement of watermarks as an additional reactive control provides complementary benefits.

Reactive controls do not actively prevent misappropriation or data transfer from happening. However, the benefits of reactive controls are multifold. To support

recovery of losses,
controls provide an
benefit to their fore
that a copy can be
Reactive controls r
Furthermore, valuab
trols, providing mar
channels being utili
and may be used co
Watermarks are
watermarks are ma
intrinsically embedd
of the copy of the d
tion of the content,
worth. As the copy
workflow, there are
retains its same perc
new format via wra
the incorporation of
ous processes conti
or technology. Mor
rather than being st
encryption or wrap
and the only protect
can be designed to
analog domains for
analysis of data tha

1.6.2 Traceabilit

Watermarks, being
to problems with w
control, watermarks
mapping transactio
are presently deplo
cation of the media
upgradeability. In a
rable to copy prote
artwork that distort
posite art. When al
and reactive contro
make money diffic

recovery of losses, so-called "tracing traitors" or "identifying pirates," reactive controls provide an audit trail for actuarial or forensic analysis. As an ancillary benefit to their forensic capability, reactive controls act as a deterrent. Knowing that a copy can be traced back to a pirate is common in traditional commerce. Reactive controls may also assist with authentication or indicate tampering. Furthermore, valuable actuarial information may be obtained through reactive controls, providing marketing information intrinsic to the data objects or distribution channels being utilized. Reactive controls are complementary to active controls and may be used concurrently.

Watermarks are a "reactive" DRM control technique. Unlike wrappers, watermarks are maintained throughout the data workflow. As watermarks are intrinsically embedded into the content, they cannot be removed during processing of the copy of the digital media. Ideally, attempts at removal result in degradation of the content and a corresponding devaluation of the content's economic worth. As the copy of the media is moved through its expected and unexpected workflow, there are no stages requiring removal of the watermark as the media retains its same perceptual qualities. As watermarks do not modify the copy to a new format via wrapping or encryption, processing and workflow used prior to the incorporation of watermarks in the media do not require modification. Previous processes continue to stay the same without the incorporation of new steps or technology. Moreover, the watermark is retained in each step of the workflow rather than being stripped off as is required in many security controls employing encryption or wrappers. Once the wrappers are stripped off, they are ineffective, and the only protection mechanism remaining is the reactive controls. Watermarks can be designed to survive format and data transformations between digital and analog domains for varying degrees of persistence. This persistence assists with analysis of data that exists in different formats or channels.

### 1.6.2 Traceability and Active Controls

Watermarks, being a part of the media rather than external to it, are not susceptible to problems with wrappers. Moreover, when used in conjunction with an active control, watermarks are not removed during the unwrapping process. By indelibly mapping transaction information to the characteristics of the media, watermarks are presently deployed in several active control environments to manage authentication of the media and enable such features as copy management and even system upgradeability. In a manner parallel to physical money, active controls are comparable to copy protection features, including ink type, paper stock, fiber, angles of artwork that distort in photocopier machines, inserted magnetic strips, and composite art. When all of these security features are reduced to digital data, active and reactive controls can be similarly compared. These controls are intended to make money difficult to reproduce, while the serial number is intended to enable audits of specific transactions. Responsibility over individual media copies via watermarks can be used to enable policies regarding use limitations, first and third party transfers, and any number of active controls.

Though active controls provide a first line of defense, they have many inherent deficiencies. By the very nature of a wrapper, it must be unwrapped to use. Similar to a crab moving out of its shell, at the point of unwrapping the media has no effective protection mechanism. In practice, several technologies have been used to actively protect the media, including physical protection. However, these additional controls have limited effectiveness given the sophistication of hackers, complexity of the wrapper, and inconveniences presented to users. Once hacks have been successfully made, it is relatively easy for less sophisticated users to deploy the same hack with little effort. Wrappers increase overall processing requirements depending on operating systems or file formats limiting persistent protection. Inconvenience is the most significant problem for the users of the media. Unless each step of the workflow is able to unwrap "securely," the process leaves exposed media vulnerable. Active controls limit the movement of information, as each process requires the unwrapping technology associated with it.

### 1.6.3 Binding Transactions, Not Just a Handshake

The placement of transactional information directly into media works has many benefits. First and foremost, it creates an audit trail embedded directly into the work. This information can include time, place, and the identities of the transferring party and the transferee of the electronic media. Whereas system logs on computers can state prior actions that have taken place on a server, these logs cannot be used to analyze two copies of the same media and state the past history of the works. Yet today, it is not uncommon that multiple copies of the same media are transferred to multiple parties, including internal and external parties. System logs are insufficient to determine cause during a forensic analysis of media discovered at an unauthorized location unless each copy is serialized. System logs also make analysis of first and third party responsibility an unsupported process, if applied alone. In practice, a unique serial or transaction number, rather than the actual, copyable information, is placed as a search index to map back to additional transaction information (e.g., name, date, time, distribution channel, transaction id, etc.) stored in a database. Such hierarchy, or layering of "unique digitized data," is beneficial for workflow separation [13] and assigning responsibility over data as it moves within and beyond an organization's electronic systems.

As a single work (or other electronic media) may be digitally copied into multiple digital works at little or no marginal cost, digital watermarks ensure that each digital work is uniquely serialized. Similar to physical money with serial numbers, each

unit is unequivoc
the same source.
trails of digital da
instance, Person
watermark "A". I
imprints a waterm
can be repeated f
transactional info
into the work via
unique yet perce
embedded audit
been transferred
a work sent to "
sent to "C." As
exact copies. Bec
embedded audit
the watermark co
is perceptually t
unique. A copy v
for transfer from
from "A."

1.7 NOW, TH

Looking backw
simpler than pro
definitive "last
ogy landscape.
paradigms or b
viewpoints are
in seeking prof
starting points,
especially with
is impacted by
erty will be su
economy, besid
are needed in f
knew what the
distribution ch
valuable works

unit is unequivocally different and perceptually equivalent from other copies of the same source. Properly deployed, digital watermarks enable inherent audit trails of digital data in any number of electronic transactions or workflows. For instance, Person A has a copyrighted work with their identity embedded as the watermark "A". In transferring a copy of the digital work to Person B, Person A imprints a watermark with identity "B" into a new copy of the work. This process can be repeated from Person B to Person C and so forth. Similarly, additional transactional information or a unique serial or transaction number may be placed into the work via a watermark. In the process, each electronic copy is digitally unique yet perceptually the same. Hence, each copy incorporates an internally embedded audit trail of its transactional history. The same work may also have been transferred by the same person to two different entities. In this scenario, a work sent to "B" is uniquely different, but perceptually equivalent to a work sent to "C." As the data is digital rather than physical, a recipient may create exact copies. Because of the watermark, each new copy must contain the previous embedded audit trail relating to its past history. Each work, independent of what the watermark contains and the number of watermarks incorporated into the copy, is perceptually the same. From an auditing and forensic point of view, these are unique. A copy with watermark "A, B" relates to a work that was last authorized for transfer from Person A to Person B and was not obtained directly from "C" or from "A."

## 1.7 NOW, THE FUTURE

Looking backward at the progress of technology, as with any hindsight, is much simpler than projecting forward. The concepts discussed here do not represent the definitive "last word," but an introduction to an important aspect of the technology landscape. DRM is a subject with so many competing stakeholders that new paradigms or business models do not necessarily appear obvious [14], and the viewpoints are not mutually exclusive. However, business is primarily an exercise in seeking profits. Measuring profitability or even accountability are invaluable starting points, but by no means is money the only perspective nor should it be, especially with regards to copyright. It is not just copyrighted multimedia that is impacted by advances and debates over DRM. Arguably, all intellectual property will be subjected to similar pressures. A valuable and fungible asset in the economy, besides time, is trust. Trust itself shapes many of the compromises that are needed in further commercializing networks [15]. An important aside: if we knew what the "blockbusters" would be, we would forgo the agents, promotion, distribution channels, specialty retailers, and all other middlemen and offer the valuable works from the back of our cars. *Caveat emptor*.

## ACKNOWLEDGMENTS

Thanks for all of the rich insight and valuable comments received over the past decade. Special thanks goes to Yair Frankel and my family.

## REFERENCES

[1] Lotus Development Corporation v. Borland International, Inc., 49 F. 3d 807, 818 (1st Cir. 1995).

[2] P. Durdik. Reverse Engineering As A Fair Use Defense To Software Copyright Infringement. Jurimetrics J., 34:451–470, Summer 1994.

[3] C. Miller. New Technology and Old Protection: The Case for Resale Royalties on the Retail Sale of Used CDs. Hastings Law Journal, 46:217–241, November 1994.

[4] Originally, T. Carroll. A Frequently Asked Questions Document Discussing Copyright Law ftp://ftp.aimnet.com/pub/users/carroll/law/copyright/faq/part2. Updated on September 11, 2002, http://www.tjc.com/copyright/FAQ/.

[5] Metro-Goldwyn-Mayer Studios, Inc. v. Grokster, Ltd. (04-480) 380 F. 3d 1154 (Sup Ct. 2005).

[6] Basically, a chip for encryption with a backdoor for the government.

[7] S. Moskowitz. *Bandwidth as Currency*, IEEE MultiMedia, pp. 14–21, January–March 2003.

[8] Barak, Goldreich, et al. *On the (Im)possibility of Obfuscating Programs*. An extended abstract appeared in *CRYPTO 2001*, pp. 1–43, August 16, 2001.

[9] R.J. Anderson. Cryptography and Competition Policy—Issues with 'Trusted Computing,' 2003 Wenk Lecture related to R.J. Anderson. TCPA/Palladium FAQ, at http://www.cl.cam.ac.uk/users/rja14/tcpa-faq.html/.

[10] Method for Stega-Cipher Protection of Computer Code, U.S. Patent No. 5,745,569, Patent and Trademark Office, 1998.

[11] There are many forms of digital watermarks. We generically use watermark to mean forensic or traceable watermark.

[12] "Active DRM" is our preferred terminology to distinguish between DRM technologies providing active controls and DRM technologies providing reactive controls.

[13] Workflow separation refers to the steps, or identifiable points, data moves as it is being prepared or processed.

[14] W. Fisher. Promises to Keep: Technology, Law, and the Future of Entertainment, PTKIntroduction.doc Draft, pp. 1–21, March 22, 2003. Also see http://www.harvard.edu/Academic_Affairs/coursepages/tfisher/Music.html/

[15] A. Odlyzko. Privacy, Economics, and Price Discrimination on the Internet [Extended Abstract], http://www.dtc.umn.edu/~odlyzko, pp. 1–16, July 27, 2003.

# 2

# Dig
# Ma

Marina

## 2.1 INTRODUC
## MANAGEM

In recent years ther
content from analo
tion from vinyl rec
formats; from book
MPEG-2, HD-TV,
Video format and
enhanced end-user
distributions, as we

Content owners
source of revenue
files containing co
an "implicit" form
specter of unlimite
compensation to th

Digital Rights
distribution of dig
technology doesn'
but instead provid
words, DRM syste
but the existence
protected.

## This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

## BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

☐ BLACK BORDERS

☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES

☐ FADED TEXT OR DRAWING

☑ BLURRED OR ILLEGIBLE TEXT OR DRAWING

☐ SKEWED/SLANTED IMAGES

☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS

☐ GRAY SCALE DOCUMENTS

☑ LINES OR MARKS ON ORIGINAL DOCUMENT

☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY

☐ OTHER: ______________________________

**IMAGES ARE BEST AVAILABLE COPY.**
**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**

# Towards a secure and de-centralized digital watermarking infrastructure for the protection of intellectual property

Philipp Tomsich and Stefan Katzenbeisser

Institute of Software Technology, Vienna University of Technology
Favoritenstraße 9–11/188, A–1040 Wien, Austria
phil@ifs.tuwien.ac.at, skatzenbeisser@acm.org

**Abstract.** The advent of the Web, electronic commerce and the creation of electronic distribution channels for content have brought new challenges regarding the protection of intellectual property. As it has become increasingly difficult to protect the distribution medium against copying, techniques for asserting the copyright on information have gained in importance. A particularly promising method is the use of digital watermarking to embed additional copyright information within data. However, central servers or certification authorities are required by most current watermarking protocols, thus limiting the wide-spread application of watermarking in electronic commerce applications.

We propose a secure, distributed watermarking scheme using trusted, tamper-proof hardware. The protocols presented provide support for copyright protection and fingerprinting in a de-centralized fashion. Extensive use of a public-key infrastructure permits the secure exchange of secret keys between trusted devices. The unencrypted, private keys never leave the hardware, rendering them unrecoverable. If adopted, this allows for the establishment of a ubiquitous digital watermarking infrastructure to support and foster e-commerce applications.

## 1 Introduction

With the increasing availability and distribution of media in a digital form, the protection of intellectual property faces new challenges. The possibility to easily and cheaply reproduce content without a loss of quality is undermining the film, music and entertainment industries. As a consequence, the question of how to effectively protect the copyright holder's interests are critical to a wide-spread acceptance of e-commerce in these application domains.

Two fundamentally different approaches exist to counteract the increased risk of copyright infringements:

- **Copy protection** attempts to find ways which limit the access to copyrighted material and/or inhibit the copy process itself. Examples include encrypted digital TV broadcast, access controls to copyrighted software through the use of license servers and technical copy protection mechanisms on the media. However, copy protection is very difficult to achieve in open systems, as recent developments for DVD [2, 11] show.

- **Copyright protection** uses embedded information to encode the copyright owner's identity within the content. Whenever the copyright of a digital document is disputed, this copyright information can be extracted to identify the rightful owner. It is also possible to fingerprint digital content with the identity of its buyer to provide for the tracing of any authorized copies. The most prominent way of embedding information in digital media is the use of digital watermarking [7].

Basically a watermarking scheme consists of two algorithms, one embedding and one extraction algorithm. The embedding algorithm inserts a watermark into digital media using a secret key, thereby generating the watermarked media. Depending on the nature of the extraction algorithm, two types of watermarking schemes can be identified. The extraction process of *private watermarking* systems takes the watermarked media, the original media, the watermark and the secret key and outputs TRUE if the watermark is actually present in. In the case of *blind watermarking* systems, the extractor extracts the watermark given only the watermarked media and the key. Watermark extraction should also be possible in the case small modifications have been applied to the marked media, i.e. the embedding process should be *robust*. Such modifications can be the result of intentional attacks in order to remove the mark or the result of coding schemes (e.g. lossy compression) and errors during the transmission [7]. Whereas secure copy protection mechanisms in open systems seems to be difficult to achieve, copyright protection systems based on watermarks and strong cryptography are feasible. As a result, considerable interest in digital watermaking exists for electronic commerce applications. However, watermarking protocols are yet to experience wide-spread use. Most approaches proposed so far, either make use of a central server to ensure the uniqueness of watermarks [1], require the generation of disjunct keys for every transaction or require the publication of private keys during the dispute resolution [6].

We propose a secure, distributed watermarking scheme using trusted, tamper-proof hardware; however, we this protocol does not attempt to restrict the use of copyrighted material and leaves this to higher level solutions. The protocols presented provide support for copyright protection and fingerprinting in a de-centralized fashion. Extensive use of a public-key infrastructure permits the secure exchange of secret keys between trusted devices. Trust is established using public-key certification and verification. The unencrypted, private keys never leave the hardware, rendering them unrecoverable. Such a distributed solution appears favorable to one based on the availability of a central server, as it is immune to denial-of-service attacks, offers far superior scalability and no-one except for the owner ever has access to the original, unmarked work.

The remainder of this paper is organized as follows: Section 2 discusses the basic requirements and problems of watermarking for copyright protection. The properties of tamper-proof hardware are summarized in section 3 and a protocol using such hardware is given in section 4. An extension to fingerprinting is presented in section 5. Some conclusions are given in section 6.

## 2 Considerations for successful watermarking protocols

Many researchers see watermarks as a cure-all solution to copyright protection problems. However, this protocol is seriously flawed and susceptible to different attacks as depicted in the following scenarios featuring three imaginary person Alice, Bob and Carol, where Alice is the rightful copyright owner, Bob commits an infringement and Carol act as judge or arbitrator:

- **Invertibility.** Bob could try to insert his own watermark in the copy he received and claim the ownership of the newly marked object. One could argue that such an attack has to fail, since Alice's original contained only her watermark whereas Bob's fake original contains both Alice's and Bob's mark, thereby clearly establishing an order of watermark insertion. However, Craver et al. [4] showed that it is—under specific circumstances—possible by an attacker to insert a second watermark in the already marked object in a way that his new mark seems to be present in the copy Alice locked away (although the attacker has no knowledge of the unmarked data). In such "invertibility attacks", Bob "subtracts" (rather than adds) his own watermark from the watermarked data, and claims the result to be the original work. The watermark detector will now detect Bob's watermark in the original object. There is no way to resolve copyright ownership in this case, as it becomes impossible to determine which object is the original.
  In order to prevent such an attack, one has to use *noninvertible marking schemes*, based on hash functions. Another possibility would be the use of a central time-stamping device.
- **Public versus private information.** When Alice is asked to prove the ownership of her works in front of a judge, she has to reveal her private key. Since in most watermarking schemes the key is coupled with the location of the watermark in the digital media, it is then possible to remove the mark once the key is public. Thus, once Alice is asked to prove the ownership of a work, *all other marks generated by the same key are removable*.
  Thus, an asymmetric watermarking system (similar to asymmetric cryptography) would be preferable, where a mark is inserted by a private key but can be checked by a public key. Unfortunately, such schemes do not exist yet. Recently Craver [6] proposed the use of zero-knowledge proofs in watermarking.
- **Buyer/Seller conflict.** Suppose the digital work is not sold by Alice directly, but by a distributor named Carol. Bob can always claim that it was actually Carol who circulates illegal copies containing watermarks identifying Bob as the customer and that he is actually innocent [9]. There is no way for Carol to prove the opposite.
- **Copy attack.** Recently Kutter et al. [8] showed that in some systems a third party is able to copy a watermark from a marked image to another (unmarked) image.

For these reasons a simple watermarking protocol in this section cannot solve the problem of copyright protection. However, a number of general requirements for practical copyright protection protocols can be established.

**Secrecy Criterion:** Watermark verification must be possible *without* revealing the secret key of the owner.

**Rightful Ownership Criterion:** A watermark should uniquely identify the owner of a digital document; thus, it should not be possible to forge watermarks (or copy watermarks between images).

**Noninvertibility Criterion:** It must at least be feasible to reconstruct a strict order of watermark insertion. Thus, if marks $W_1, \ldots, W_n$ are found in the document, it must be possible to determine which mark was the first one (thereby preventing "invertibility attacks").

**Decentralization Criterion:** The copyright protocol should not rely on central infrastructure, but rather on the existing public-key infrastructure.

## 3 Tamper-proof hardware

Software executing within the main memory of a computer is always susceptible to manipulation of observation, weakening the secrecy criterion. An intruder may be able to retrieve secret key information from main memory during the execution of an encryption algorithm. Dedicated hardware may provide a far better protection against attackers by limiting the number of access points. In combination with audited protocols and certified software, attackers can be locked out of the system. During the last few years a number of *tamper-proof* hardware devices have been developed and deployed, the most popular being smart cards used for identification purposes and financial transactions.

The physical security of information stored in tamper-proof hardware usually starts with the combination of computer memory and processor in a single package with a protective enclosure. Given the outer enclosure sufficiently protects such a device against unauthorized access the the chip, it is extremely difficult to examine the contents of the memory cells within the chip. It is also difficult to to intercept the electrical signals passing between the processor and memory. All access to the hardware is carried out using dedicated access points and protocols. The hardware runs a tiny operating system, which implements the communication protocol and provides security. The access to state information of processes running within the tamper-proof hardware thus requires fairly expensive equipment and unhampered access to the hardware under attack.

In order to ensure that the software executing within the tamper-proof hardware is trustworthy, only manual inspection and certification is possible. However, using a read-only memory to store the program code, the hardware remains trustworthy for its entire lifetime. If the distribution of secret information is limited to certified devices, no danger of a public disclosure exists. If a public-key infrastructure is to be built on such devices, the public keys may be signed by a well-known certification authority (e.g. an international standardization organization). Although seemingly centralized, the central authority is only needed prior to the deployment of the hardware—the operation does not require access to any central site or service. Since software updates will likely be necessary, appropriate mechanisms to ensure the integrity and trustworthiness of the new software packages have to be added to updateable hardware. Such a software update infrastructure can be built using public-key cryptography and one-way hash functions.

## 4 Copyright protection protocol based on tamper-proof hardware

We present a copyright protection protocol which is based on tamper-proof hardware and a traditional public-key infrastructure. It is assumed that every user has access to tamper-proof hardware which contains the public key $E_{CA}$ of a certification authority and a certified public/private key pair $E_{H_A}/D_{H_A}$. It is assumed, that every user has a key pair $E_A/D_A$ as part of an infrastructure for legally binding digital signatures. The protocols presented are assumed to be implemented in tamper-proof hardware. Copyright protection is based on four protocols: watermark key generation, watermark insertion, watermark extraction and a dispute resolution protocol.

### 4.1 Watermark key generation

The purpose of this protocol is to generate a "watermark key envelope", which is used in the following protocols. Basically, a watermark key envelope consists of an encrypted random watermark and a string describing the identity of a user, signed by the certification authority.

1. Alice requests the signed public key of the hardware and verifies whether it is signed by an agreed authority which ensures a conforming software within the trusted hardware.
2. Alice sends an encrypted request to her hardware; this request contains the public key $E_A$ of Alice along with a string of her identity $Id$, signed with her private key (we denote this by $D_A(Id)$). Furthermore Alice sends a certificate of her public key $D_{CA}(E_A, Id)$, i.e. her public key and her identify string signed by a certification authority.
3. The hardware generates a random watermark key $K$ and encrypts $K$ with its public key $E_{H_A}$, yielding $E_{H_A}(K)$ and constructs the watermarking key envelope, consisting of the encrypted watermark $E_{H_A}(K)$, the signed user identification $D_A(Id)$ and the certificate $D_{CA}(E_A, Id)$ received:
$$W_K = \langle E_{H_A}(K), D_{CA}(E_A, Id), D_A(Id) \rangle .$$
The hardware signs the envelope and returns it to the user.
4. Alice checks the signature on $W_K$ to ensure unmodified transmission and stores $W_K$ for future use.

The watermark key envelope provides a secure transfer and storage medium for the secret watermark key and related information identifying the key holder. This ensures the *secrecy criterion*, because the unencrypted key never leaves the hardware. Even the key holder cannot access or manipulate the information contained without using the trusted hardware. This guarantees that only valid operations can be performed. The contained user identification allows the watermark verification process to uniquely identify the key holder (a prerequisite for the *rightful-ownership criterion*) within the limits of current digital signature standards. Signing the key envelope with the secret hardware key ensures that an intruder cannot insert a pre-fabricated watermark envelope, as no direct verification of the envelope's content is possible for the user.

It should be noted that one user can have multiple watermark keys. Furthermore, one user can also have more than one piece of hardware; one hardware device can also be shared between different people.

### 4.2 Watermark insertion

This protocol inserts a watermark in a digital object $O$, thereby using a noninvertible watermarking scheme. We assume a method similar to Craver et al. [5] or Qiao and Nahrstedt [10]. A noninvertible scheme is based on a hash of the original data. Suppose the watermark consists of $n$ watermark bits $w_0, \dots, w_n$ and the first $n$ bits of a hash of $O$ are $b_0, \dots, b_n$. Depending on the value of $b_i$, the watermarking algorithm chooses among two possible ways of inserting the watermark bit $w_i$. Assuming a "perfect" hash function $H$ (i.e. a hash function which hashes even perceptually similar images to completely different bit-strings), it is believed that such watermarking schemes are not susceptible to inversion attacks: suppose an attacker wants to "subtract" a fake watermark $W'$ from an already watermarked data $O'$, thereby creating the fake original $O''$. Since in noninvertible marking schemes the location of $W'$ depends of the fake original $O''$ which is not yet known, the attacker has to guess a bit-string $b_1, \dots, b_n$ and a mark $W'$ in a way that when $W'$ is subtracted from $O'$, the result hashes to $b_1, \dots, b_n$. This should not be possible when using a one-way hash.

1. Alice requests the hardware public key and verifies it.
2. Alice sends the original data $O$, a string *Desc* describing $O$ and a previously generated watermarking key $W_K$ back to the hardware. All data must be encrypted using the hardware public key to fend off any attackers intercepting transfers to the trusted hardware.
3. The hardware extracts the encrypted random watermark $E_{H_A}(K)$ out of $W_K$, decrypts it to obtain the watermarking key $K$. It then watermarks $O$ using a noninvertible scheme and watermark key $K$; the watermark itself should consist of a string describing Alice's identity.
4. The hardware sends the watermarked image back to Alice, along with verification token consisting of all information necessary to verify the watermark. This includes the description, the hardware public key, the watermarking key envelope and a one-way hash of the original object used by the non-invertible marking scheme:
$$Ref = \langle Desc, E_{H_A}, W_K, H(O) \rangle .$$
The hardware signs *Ref*, returns it to Alice and clears its memory.
5. Alice retrieves the marked data and stores *Ref* for use in a watermark verification protocol or a dispute resolution protocol.

Using a non-invertible watermarking scheme ensures the satisfaction of the *non-invertibility criterion*. During the process, a verification token is generated, which encapsulates all the information necessary in the verification and dispute resolution protocols. The verification token returned to the user, contains a watermarking envelope with an encrypted key which can only be decrypted by the original watermarking hardware. In order to use it with a different hardware device, it needs to be decrypted by the original hardware and encrypted for the new hardware device. This is necessary to uphold the *secrecy criterion*.

### 4.3 Watermark verification

The watermark verification protocol is straightforward to implement. The same hardware that was originally used to watermark the data is given a marked media and the verification token. It then verifies the presence of the watermark in the media, using information from the verification token.

1. Alice requests the public key from the hardware and checks whether the hardware is trustworthy.
2. Alice transfers the marked object $O$ and the associated verification token *Ref* into the hardware. The transferred data is encrypted using the hardware public key in order to prevent attackers from inserting fake data.
3. The hardware checks the signature of the verification token, extracts the hash value $H(O)$ and decrypts the random watermark key $K$ contained in the watermark key envelope $W_K$. After this process, the hardware checks whether Alice's watermark is contained in $O$, thereby using the watermark key $K$ and the hash $H(O)$, and returns the answer TRUE or FALSE. It then clears its memory.
4. If the answer of the hardware was TRUE, it supports Alice's claim that bob infringes her copyright.

Note that there is no need for the hardware to check the identity of its user, which conforms with the *decentralization criterion* as no central directory is necessary. If this identity is in question, authentication will be performed during a dispute resolution protocol. Bob may now confess that he was actually illegally distributing Alice's media. Otherwise, Alice will start a dispute resolution protocol in front of an arbitrator. This arbitrator will again verify the watermark and query the certification authority for the validity of Alice's public keys. The *secrecy criterion* holds for watermark key, as it never leaves the hardware unencrypted.

### 4.4 Dispute resolution protocol

Probably the most difficult protocol is the dispute resolution protocol. This protocol involves three parties: Alice, Bob and an arbitrator/judge Carol. Basically, Carol will verify the watermark in *her* hardware, thereby preventing possible allegations by Bob that Alice is actually cheating in the verification process. For this purpose, the judge asks Alice's hardware to provide a verification token that is suitable for her hardware (i.e. re-encrypt the verification token). Carol's hardware can now verify the validity of the mark and check the identity of Alice.

In fact, the dispute resolution protocol does not attempt to determine the actual holder of the copyright, but rather establishes an strict precedence order on the claims, similar in spirit to the ordering system used for patent rights. The actual copyright holder can only be determined, if he/she participates in the protocol.

1. Alice transmits the public key of her hardware to Carol.
2. Carol verifies that Alice's hardware is trustworthy.
3. Carol then asks Alice's hardware to recode the verification token for her hardware and provides her hardware public key to Alice.
4. Alice verifies Carol's hardware key to determine whether the hardware is trustworthy. If this succeeds, both parties have established that their hardware may communicate using the provided keys.

5. Carol's hardware now receives the verification token, recoded and encrypted to her hardware. The recoding process involves the decryption and re-encryption of the secret watermarking key (contained in the watermarking key envelope). The second layer of encryption ensures that the data can not be manipulated during the transmission. Additionally, the sending hardware signs the token with its private key to uniquely establish the originator.
   In more detail, Alice's hardware receives *Ref*, extracts and decrypts the contained information and returns the token
   $$\langle Desc, E_{H_A}, H(O), E_{H_C}(K), D_{CA}(E_A, Id), D_A(Id)\rangle$$
   where $E_{H_C}$ denotes the public key of Carol's hardware. *Desc* is a string describing the digital data, $E_{H_A}$ is the public key of Alice's hardware, $E_{H_C}(K)$ is the random watermark encrypted with Carol's public hardware key, $D_{CA}(E_A, Id)$ is a certificate of Alice's public key and $D_A(Id)$ is Alice's signed identity. The entire token is signed by Alice's hardware using the secret hardware key $D_{H_A}$.
6. Carol's device checks the signature on the token received, extracts the necessary information. Then, the device decrypts the random watermark key contained in the watermarking key envelope and verifies the presence of the watermark using the hash. Once the watermark is accepted as genuine, it remains to control the identity of Alice to detect the man-in-the-middle: Carol's hardware checks the signature on the certificate $D_{CA}(E_A, Id)$ using the public key of the certification authority and uses the private key $E_A$ to verify the signature $D_A(Id)$. However, it remains to verify whether the person identified by $Id$ is actually the communication partner expected. Existing infrastructure for legally binding digital signatures can be used in this phase.
7. If all tests passed and Alice's watermark is indeed present, Carol's hardware outputs TRUE.

In a simple case, Bob may now confess that he has actually stolen Alice's data. However, Bob could also claim that he is the rightful owner and that Alice has actually stolen his image and inserted her watermark into it. Carol has to resolve this case by checking the presence of watermarks in the digital data Alice and Bob claim to be the originals. We can distinguish four cases:

- *Bob's original contains Alice's mark but Alice's original does not contain Bob's mark*: in this case, Bob clearly inserted his mark after Alice. The court may conclude that Alice is the rightful owner.
- *Alice's original contains Bob's mark but Bob's original does not contain Alice's mark*: this case is similar to the last one; clearly, Alice inserted her mark after Bob and so the court may conclude that Bob is the rightful owner.
- *Both Alice's and Bob's original contain no detectable watermarks*: in this case, no conclusion can be drawn; either Bob or Alice got an unmarked version of the image owned by the other one or both Alice and Bob independently inserted their own watermarks into an image actually owned by a third person. This third person may have watermarked the image or not.
- *Both Alice's and Bob's original contain both watermarks*: this is the classical *deadlock situation* produced by inversion attacks. Again, no conclusion can be drawn.

In the first two cases, it was possible to resolve the copyright situation; in the last two cases a final conclusion cannot be drawn and the dispute must be settled in a traditional court case.

Even the first two cases are more problematic than they may seem: since the dispute resolution protocol is only a three-party protocol, there might by the possibility that both Alice and Bob have actually stolen the image from another party which does not participate in the protocol. In this case, the claimed originals might contain other watermarks. Since the watermark key of this unknown party would be required to verify that assumption, Carol is not able to exclude this possibility until she has checked all watermark keys from *all possible parties*, which is obviously not feasible.

Obviously the third and fourth cases are most problematic. One could argue that the fourth situation does not happen when using noninvertible watermarking systems. The third case should never happen in reality either, as it always results from neither party having inserted a watermark or from uncontrolled access to the unmarked original, which is then copied by the infringer.

## 5 Fingerprinting protocol

The protocols presented in the last section do not allow the tracing of users selling illegal copies of digital data. In addition to the normal watermarking functionality, it is required that a mark should identify the buyer of the digital object uniquely. No customer should be able to falsely deny that he distributed illegal copies. The fingerprinted media should only be known to the customer to avoid false claims of infringement.

It is straightforward to add such functionalities. However, the marking algorithm has to be modified to avoid *collusion attacks*; assume that several copies of one digital object are sold and that an attacker has access to $n$ such copies. By comparing the copies, he can find least some of the modifications applied during the marking process and try to remove them. To elude this attack, the watermark is encoded prior to the embedding process in such a way that several watermarks have a common intersection which cannot be found by comparison. Boney and Shaw presented an encoding for this purpose in [3]. In order to avoid a buyer/seller conflict, the marked data must not be known to the merchant. This can be provided for in two ways: the sold data can either be marked in the buyer's media or the marked media must leave the merchant's hardware encrypted. The watermark insertion protocol can be modified accordingly.

## 6 Conclusions and future research

We argue that watermarking alone is not sufficient to resolve rightful ownership of digital data; a protocol relying on the existing public-key infrastructure (which is also used for digital signatures) is necessary. It seems that the primary vulnerability of the presented protocol is the watermarking algorithm itself; most known watermarking systems are sensitive to intentional distortions of the digital data and do not merge the digital data and the watermark completely, as copy attacks show. For these reasons, the software used for watermarking will have to be updated regularly. A secure distribution protocol will become necessary to support these updates. Additionally, the presented solution poses open problems, if hardware is rendered inaccessible as the hardware's secret keys are otherwise compromised.

The protocol presented in the previous sections eliminates the problem of revealing the private key in front of a judge when verifying a watermark. A distributed solution overcomes the main disadvantages of a central solution: limited scalability, a single point of failure and the dependence on the trustworthiness of the service provider. Furthermore, extensive use of public key cryptography assures the secure exchange of keys and renders man-in-the-middle attacks very difficult; trusted tamper-proof hardware is used to conceal the actual watermarking operation. It is easy to imagine that a specialized microchip—which could integrate other functionality used to support secure e-commerce, such as secure electronic transactions and public-key cryptography—can be cheaply produced, given the number of potential customers. When a robust and non-invertible watermarking system is used as a building block for the proposed protocol, we believe that this protocol allows the establishment of a sufficiently secure digital watermarking infrastructure to support and foster e-commerce applications.

## References

1. A. Adelsbach, B. Pfitzmann, A.-R. Sadeghi, "Proving Ownership of Digital Content" in *Proc. of the Third Intl. Workshop on Information Hiding*, LNCS 1768, 2000, pp. 117–133.
2. J. A. Bloom, I. J. Cox, et. al., "Copy Protection for DVD Video", in *Proc. of the IEEE*, vol. 87, no. 7, July 1999, pp. 1267–1276.
3. D. Boneh, J. Shaw, "Collusion-Secure Fingerprinting for Digital Data", in *Proc. of the CRYPTO'95*, LNCS 963, 1995, pp. 452–465.
4. S. Craver, N. Memon, B. L. Yeo, M. M. Yeung, "Can invisible watermarks resolve rightful ownership?", in *Proc. of the SPIE 3022, Storage and Retrieval for Image and Video Databases*, 1997, pp. 310–321.
5. S. Craver, N. Memon, B. L. Yeo, M. M. Yeung, "Resolving Rightful Ownerships with Invisible Watermarking Techniques: Limitations, Attacks and Implications", in *IEEE Journal on Selected Areas in Communications*, vol. 16. no. 4, 1998, pp. 573–586.
6. S. Craver, "Zero Knowledge Watermark Detection", in *Proc. of the Third International Workshop on Information Hiding*, LNCS 1768, 2000, pp. 101–116.
7. S. Katzenbeisser, F.A.P. Petitcolas (eds.), *Information Hiding Techniques for Steganography and Digital Watermarking*, Boston, London: Artech House, 2000.
8. M. Kutter, S. Voloshynovskiy, A. Herrigel, "The Watermark Copy Attack" in *Proc. of the SPIE 3971, Security and Watermarking of Multimedia Contents II*, 2000.
9. N. Memon, P. W. Wong, "Buyer-seller watermarking protocol based on amplitude modulation and the El Gamal Public Key Crypto System", in *Proceedings of the SPIE 3657, Security and Watermarking of Multimedia Contents*, 1999, pp. 289–294.
10. L. Qiao, K. Nahrstedt, *Watermarking Schemes and Protocols For Protecting Rightful Ownerships and Customer's Rights*, Research report, Dept. of Computer Science, University of Illinois at Urbana-Champaign, 1997.
11. B. Schneier, *DVD Encryption Broken*, Crypto-Gram Newsletter 11/1999, available at `http://www.counterpane.com/crypto-gram-9911.html`.

# What is Acceptable Quality in the Application of Digital Watermarking: Trade-offs of Security, Robustness and Quality

Scott Moskowitz,
Blue Spike, Inc. 16711 Collins Avenue No. 2505, Miami Beach, Florida 33160
scott@bluespike.com

## Abstract

*Quality is subjective. Quality can be objectified by the industry standards process represented by such consumer items as compact disc ("CD") and digital versatile disc ("DVD"). What is lacking is a means for not only associating the creation of valued intangible assets and extensions of recognition but establishing responsibility for copies that may be digitized or pass through a digital domain. Digital watermarking exists at a convergence point between piracy and privacy. Watermarks serve as a receipt for information commerce. There is not likely to be a single digital watermark encoding scheme that best handles the trade-offs between security, robustness, and quality but several architectures to handle various concerns. The most commercially useful watermarking schemes are key-based, combining cryptographic features with models of perception. Most importantly, in audio watermarking there currently exists mature technologies which have been proven to be statistically inaudible.*

## 1. Introduction

The efficacy of copyright management systems will depend largely on keeping "security" out of view from consumers while enabling clear responsibility to be attributed to the media content being traded. Consumers have repeatedly rejected access restriction and registration protocols as currently deployed in favor of open peer-to-peer systems. Meanwhile, the digital watermark research literature is littered with assertions concerning "quality" which have been made without comprehensive "golden ears" listening tests, such as those conducted for the Secure Digital Music Initiative's ("SDMI") Phase 2 standards process or similar tests that have been conducted in the visual applications field. Security and quality are complex and subjective.

Complicating matters is the inherent difficulty with implementations of digital rights management ("DRM") systems on consumer PCs that typically lack realistic provisions for authenticating digital objects. Ignoring historical precedent and legal province of "fair use" and the "first sale doctrine" serves to obscure the economic value attributed to content. In an ideal digital watermarking system, maintenance of the intended perceptible quality must be weighed against the technical reality of trade-offs with security and robustness against attack. Determining tampering or attributing responsibility for copies are inherent features of economic activity. Successful commercialization requires a focus on the perception of value; the file format must be relegated to convenience [1].

Without an audit trail, or the creation of receipts for content, a means of settling responsibility for particular digital objects will prevent successful commerce in an information economy. The general need for commercial deployment of workable digital watermarking schemes is best represented by the widespread acceptance of Napster™, and its progeny, including, Music City™, KaZaA™, et al.

The presence of a content identification watermark is the hook to facilitate commercial markets surrounding the *use* of music, and other media, by consumers. Some of these uses include: monitoring of broadcast playback by performing rights organizations ("PROs"), premium services for peer-to-peer music distribution networks (a commercial Napster), and consumer content identification services (like Gracenote™/CDDB for individual tracks). The cost on a computational resource basis is lower than competing identification systems using so-called "signal fingerprinting" and onerous application of DRMs that obscure any *a priori* willingness of general consumers to pay for content [2]. Furthermore, the cost is borne by each client in a distributed manner, avoiding processing and bandwidth bottlenecks, similar to the way that Napster distributed storage.

In this paper, a description of several of the decoding system applications, and why watermarks are a necessary feature of any workable market for the commercial exchange of content will be highlighted. Included is a comparable statistical measure of the actual maturity of audio digital watermarking having been proven to meet the most stringent, if not subjective, standards of sonic quality.

## 2. Broadcast Monitoring

At present, a variety of technologies are used to monitor the playback of sound recordings on broadcast outlets. Digital watermarking is a better alternative to all of the deployed technologies because it couples automated detection with extremely high reliability. A single PC-based monitoring station can continuously monitor up to 16 channels of audio broadcasts 24 hours a day with no human interaction. The results of the monitoring are assembled at a central server and made available to interested licensees, such as the PROs, for a fee equivalent to the price they currently pay for monitoring data. Unlike currently deployed systems, there is an extremely low statistical chance of misdetection. Additionally, the system can distinguish between otherwise identical versions of a song, which are watermarked for different distribution channels, further improving the quality of the reported data.

Deployment of such a system requires two things: a monitoring infrastructure and the watermarks to be present in the content. Leading monitoring companies have developed and deployed extensive infrastructures that have been designed to identify certain encoded audio and video signals as they are distributed. Watermarking music or video is planned by all major entertainment companies, those who possess closed networks, as well as, those involved in advertising.

## 3. Peer-to-Peer File Sharing

The immense popularity of peer-to-peer file sharing ("P2P"), in combination with recent legal rulings, presents a challenge: how to commercialize a file-sharing network. Watermark-based content identification is the solution. Each track is to be identified by the client's computer using a watermark detector. Ideally, the detector may be upgraded or replaced by a plurality of watermarking algorithms, if said algorithms are generated in combination with an upgradeable cryptographic key for such use. A so-called "steganographic cipher key" performs identification and authentication functions without revealing the unwatermarked original media content. The identity or authenticity of the track is then used to filter the server search engine, so that each subscription level only provides access to "allowed" content. Signal fingerprints or web trawlers cannot independently establish responsibility for any given digital object at comparable measures of computational overhead as embedded watermarks but can be used to reduce forensic searches for particular files.

As there are many embedding techniques and compression algorithms; so there should be support for many types of watermarking embedders and detectors. That a key-based watermark process essentially maps or concatenates a cryptographic signature in such a manner as to mimic the perceptibility of any given media object, emphasis on authenticity of digital objects is likely to assist in accurately determining what consumers are willing to pay for. These keys may also be used to watermark portions of specific areas of a signal or even save signal characteristics to the key to assist in detection or decoding watermark message data. Collectively, the ability to tamperproof or restore a suspect digital object with a watermark key is invaluable to maintain authorized information-based markets. Here is how it works in action:

### 3.1. Encoding

Encoding happens at the mastering level of each sound recording, as currently contemplated by the major label music companies as well as the major studios for video. Downstream, "transactional" watermarks are also considered. Each song is assigned a unique ID from the identifier database, and that ID is encoded in the sound recording after all other mastering processes are completed, but prior to the song being prepared for a specific distribution channel. To enhance imperceptible encoding of those few audio or video recordings that require special processing, human-assisted watermark key generation is readily available.

### 3.2. Decoding

Decoding happens each time a new song is made available on a P2P user's computer. A highly efficient background process decodes each sound recording, and queries P2P's main server as to the status of the selected track. The server would respond that the sound recording falls into one of the following categories:

**Uncontrolled:** The sound recording either does not contain a watermark, or the copyright owner has chosen to make the song freely available to all users. In this example, the sound recording will be freely available to pass through the P2P server.

**Premium:** The sound recording is part of a subscription package and is made available only to the premium subscriber of that subscription package.

**Restricted:** The sound recording is not authorized to be shared on the main server and will not be available for file sharing purposes.

## 4. A Real World Example

Alice is a Napster user. She has a hard drive directory of audio files which her Napster application monitors. She rips a new CD into that folder and starts the Napster application. The application reads the watermark on each track to identify those tracks. The new tracks, like all on her computer, are available for her own, unlimited, use.

When Alice connects to the server, her computer broadcasts the identity of all of the sound recordings in her shared folder. These, are a mix of uncontrolled, premium, and restricted content, as determined by the server at that time. For the new tracks that were recently added to her folder, the server identifies that one song is premium, and the others are uncontrolled.

Bob is a Napster user, and is looking for music. He is a premium subscriber. The Napster server makes the uncontrolled and premium music on Alice's computer available to Bob.

Carl is another Napster user, but not yet a subscriber. He sees only the uncontrolled music when he logs on to the Napster server.

This system provides minimum impact on consumers, while maintaining the safeguards necessary for the sharing of copyrighted material. Each user is not prevented from using restricted songs on their own computer, since in most cases they will have purchased them legally, for instance on CD or by subscription. Those songs are simply not available to others against the wishes of the copyright owner. No other approach to the rampant problem of unfettered file sharing is technically reasonable. When combined with technologies such as a content-specific cipher, which encrypts data in such a manner as to retain perceptibility but distort the media content in a tiered fashion (a predetermined key or key pair combined with a transfer function), copyright owners can estimate the highest optimized mix of quality thresholds demanded by consumers over a network in real time.

Users, in this scenario, purchase individualized keys (essentially tied to their public key or some equivalent digital credential for purchase options) based on observable music, video, or images, with reasonably open access that improve the quality of the music, or other media, as consumers "click through" to higher quality thresholds. A reduction in server overhead and cost, as well as maintenance of recognizable but secure media files, combined with digital watermarking, represent the state of the art in addressing file sharing. This also allows for multiple subscription levels based on content types and quality settings. The need to store multiple versions, both compressed and uncompressed, as per requirements for typical DRM systems, in an encrypted state is likewise reduced. Commercially, owners or aggregators of content will be able to estimate payment and bandwidth resources in real time. A natural extension is to provision paths of packets, that comprise media content, demanded between users, to efficiently provision bandwidth at the highest market price.

In the event that the sound recordings are not available with watermarking, application of signal recognition (fingerprinting) offers additional coverage. A unique abstract of the selected sound recording is taken and its signal characteristics are compared to an associated database. This comparison will identify the name of the performance if the sound recording is included in the database. Simple hashes or checksums of the audio file are ineffective given the range of reasonable alterations conceivable. Predetermination of the types or amount of signal manipulations expected on the audio file can be used to create a better, more robust "signal abstract" (which may be stored publicly, privately, or at a certification authority to point out authorized versions of the recording) than currently available signal fingerprinting applications. Application to other forms of media is obvious.

The signal recognition application is primarily useful for legacy, unwatermarked, material. This specifically limits the scope of the signal fingerprint database, which is crucial to maintaining the feasibility of fingerprinting. At present, no entity has demonstrated fingerprint technology that can economically scale to cover the daily increase in available media content. Nor can it be expected that "versioning" of the content in question will decrease in the future. With versioning of media content, more personalized exchange of any particular digital object is likely to require a means to independently authenticate objects without requiring predetermination of all possible manipulations of the media object in question.

## 5. Consumer Song Identification

Gracenote (formerly CDDB) offers a hugely successful system to identify physical CD's based on their Table of

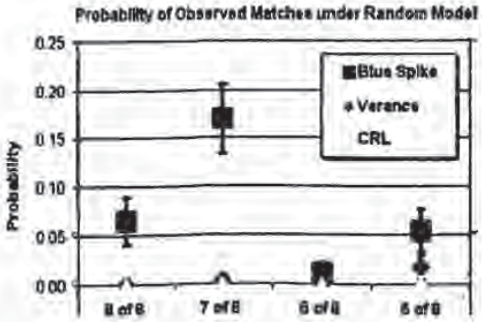
Contents. The hole in the system is that it is useless for content that arrives as an individual digital track. An MP3 found on a peer-to-peer system can arrive without any linkage to the distributor or artist. Watermarking can fix this, allowing an anonymous track to be reassociated with its creator, and facilitating sales by all of the members of the value chain.

An inexpensive watermark detector would be added as a feature or plug-in to all popular music players, just as the present Gracenote software is included. Any incoming track could be detected and then decoded, and a resulting query could be made to a server which not only identifies the track, but places it in a sales context for the up-sell of all manner of associated items, from other tracks by the same artist, to concert tickets and merchandise.

Best of all, the consumer's identification act also provides critical data on the use and popularity of each track. Here the watermark is crucial, because it can distinguish between identical tracks obtained from different sources, thus informing the viability and market potential of different modes or even channels of distribution. Finally, if the distribution channel is correctly identified, the consumer can be up-sold the appropriate items. For example, if they recorded the song from an Internet broadcast, sell them the CD.

## 6. "Audio Quality" by Statistics: SDMI

Much has been ignored or misunderstood in the research literature concerning acceptable quality parameters for digital watermarking systems. Given the generally higher sensitivity to distortion in the human auditory system, and its relevance to any psychoacoustic modeling, this paper offers opinions based on the most extensive audibility testing endured over the past six (6) years. This testing has been conducted on a number of different encoding schemes: least significant bit (LSB), adaptive quantization, amplitude masking, and several variations of mature psychoacoustic masking has yielded statistical proof that at least one audio watermarking technical is "inaudible" and technically mature. Most of this audibility testing has been conducted under confidentiality agreements with little if any provision for publicly benchmarked results. Moreover, automated watermarking systems, not the far more flexible application of key-based systems, have been exclusively emphasized for unknown reasons. The exception was the lengthy, heavily publicized, and comprehensive SDMI Phase 2 listening tests. The results presented herein were prepared by an independent doctoral statistician hired by the SDMI organization.



[Fig.1]

**Figure 1. Values above 0.05 indicate agreement with a random model. A digital watermark was less likely to have been detected. Values under 0.05 indicate disagreement with a random model. A digital watermark was more likely to have been detected [3].**

While it might be obvious that most commercially valuable music is loud and compressed enough to make any watermarking system acceptable from a sonic quality perspective, all of the significant commercial tests which have been conducted have been focused almost exclusively on classical pieces with very little data hiding space. Unfortunately, most testing has also focused on robustness without provisioning for key-based systems that can authenticate audio files and carry enough data in the key to assist in determining the original recording's scale or other signal features, without requiring the original unwatermarked file. Watermarking is a mature, flexible analog to its real world counterpart: that significant feature of commerce– the receipt. Without provably secure watermarks, or receipts, it is not likely any technology will satisfy the expectations of rights owners, consumer electronics manufacturers, information technology vendors and the public at large.

## 7. Conclusion

Consumers have created and embraced particular usage models for music, which includes CD copying, file-swapping, and format indifference. They expect to be able to play music on any of a number of device platforms, from stereos to computers to cell phones. Any system of music distribution that ignores or significantly impedes these models will meet with limited success.

More pointedly, the economics of DRM are questionable at best [4, 5]. The cost of recognition, promoting or otherwise creating demand for information content is separate from responsibility once that information content has been transacted. Access restriction threatens the viability of the historic reality that a few copyrights account for a lion's share of revenues. In 1999, for instance, only 0.03% of compact discs accounted for over a quarter of all revenues [6]. In 2000, 0.35% of all albums released accounted for over half of all revenues: 88

releases represented slightly over 25% of revenue [7]. Similar market realities apply to all forms of entertainment, including video, limiting any supposition that we can predetermine the success of any given media content release [8].

Arguments that "superdistribution" will replace market realities lack any real world examples; in fact, financial success generally boasts models seeking monopolistic or oligopolistic control of profitable intellectual property. As with physical media distribution emphasis is better placed on enabling differentiations between authorized and pirated versions of a given media content file copy or stream. Concatenating a digital signature to a media file, a key-based digital watermark, is the most appropriate means to enable markets for the open, accessible exchange of media content. Ultimately, key-based digital watermarks enable a balance to be struck between privacy and piracy. Moreover, they assist in providing transparency to replace statistical models currently relied upon by market participants. Essentially enabling receipts for information commerce. It is the conduit through which the business of music, and media in general, will be conducted, now and in the future.

## References

[1] "Deciphering Music's Digital Devolution", Billboard, Timothy White, April 28,2001, p. 10.

[2] "It's the Pricing, Stupid!", Stereophile, October 25,2001 {online edition}.

[3] "An Evaluation of the SDMI Listening Test", Eugene Ericksen, PhD, SDMI Foundation, December 1, 2000.

[4] "Music industry still in first gear online", Reuters, January 7, 2002, {http://news.cnet.com/news/0-1005-200-8395107.html}.

[5] 'Will Fixation on Security Silence the Trumpets of Fame", Digital Mogul, Scott Moskowitz and Peter Cassidy, Volume 3 Report 7 {online edition},

[6] "The Heavenly Jukebox", Atlantic Monthly, Charles C. Mann, September 2000 (online edition).

[7] "SoundScan Numbers Show .35% Of Albums Account For More Than Half Of All Units Sold", Billboard, April 28,2001, p. 66 .

[8] "Will 'Harry' have Legs?" The Wall Street Journal, November 30,2001, p. W4.

# Secure Watermark Embedding through Partial Encryption

Aweke Lemma, Stefan Katzenbeisser, Mehmet Celik, and Michiel van der Veen

Philips Research Europe
High Tech Campus 34
NL-5656 AE Eindhoven, The Netherlands
{aweke.lemma, stefan.katzenbeisser, mehmet.celik, michiel.van.der.veen}@philips.com

**Abstract.** Secure watermark embedding allows to securely embed a watermark into a piece of content at an untrusted user device without compromising the security of the watermark key, the watermark or the original. In this paper, we show how secure embedding can be achieved by using traditional watermarking schemes in conjunction with partial encryption techniques, which were primarily developed to facilitate fast encryption of media content. Based on this concept, we develop two new efficient secure embedding mechanisms, one for the MASK watermarking scheme operating on baseband audio and one for a spread spectrum watermarking scheme operating on MPEG-2 encoded video streams.

## 1 Introduction

In the past few years we have experienced a clear shift from classic content distribution channels, such as CDs or DVDs, towards electronic content distribution (ECD). Even though electronic distribution offers new business possibilities for content providers, the risk of un-authorized mass re-distribution largely limited the widespread adoption of digital distribution channels. Digital Rights Management (DRM) systems try to minimize the risk of copyright infringements by using cryptographic techniques to securely distribute content to client devices and enforce proper usage. Encryption, however, can only offer a partial solution to the problem of unauthorized distribution. Eventually, the content has to be decrypted and presented to the user in (analogue) clear-text form, from which copies can easily be made and re-distributed.

Forensic tracking watermarks [13]—which may be used in place of or in conjunction with traditional DRM/encryption methods—allow to enforce usage rights beyond the digital domain. In a forensic tracking system, each copy of the distributed content is watermarked with a unique transaction tag, which links that copy either to a particular user or to a specific device. When an unauthorized copy is found, the embedded watermark (carrying the transaction tag) uniquely identifies the source of the copy, and allows to trace the user who has re-distributed the content. Even though forensic tracking in itself does not prevent unauthorized re-distribution, the risk of being caught acts as a strong deterrent.

In current forensic tracking systems, forensic watermarks are embedded into the content directly by a trusted distribution server before the content is released onto a distribution network. This model, however, severely limits the applicability of forensic watermarks in forthcoming content distribution models:

- Integrating forensic tracking watermarks into large-scale ECD systems brings challenges with regard to security, system complexity, and bandwidth usage. As the ECD server needs to embed a unique watermark into each copy of the content, both the server load and the bandwidth requirements for content transmission scale linearly with the number of users. In large-scale content distribution applications, the watermark embedder at the server side turns out to be a major performance bottleneck. In addition, as the content is personalized for each user, distribution requires a point-to-point channel between the ECD server and the client, prohibiting the use of broadcasting, multicasting and caching, which significantly reduce the bandwidth usage for content transmission.
- In addition to the above-mentioned performance problems, server-side watermark embedding is unsuitable in forthcoming content distribution systems which employ a clear separation between content providers and license brokers. For example, in the OMA DRM model [2], content is allowed to float in a network freely in encrypted form. Once a party wishes to access the content, it purchases a license from a clearance center and obtains a decryption key. Due to the absence of a central distribution server, server-side watermark embedding is not applicable in this scenario.

These limitations could be circumvented if the untrusted client devices themselves perform watermark embedding. The major obstacle to be solved is that watermark embedders require knowledge of a secret watermarking key, which, once exposed to an attacker, allows to effectively remove watermarks. Thus, watermark embedding at the client must be done in a way which does not compromise the security of the keys; in addition, neither the watermark nor the original content should be available for the client. In the sequel, we will call client-side watermark embedding systems achieving these security properties *secure watermark embedding*. The use of secure client-side embedding can overcome both above mentioned limitations: it shifts the computational burden of watermark embedding to the client, allows to use broadcasting techniques to distribute encrypted content, and facilitates distribution models where no central server is involved in the actual purchase phase.

Secure watermark embedding transmits to the client an encrypted version of the original content together with some helper data, which implicitly encodes the watermark to be embedded. The client can use this personalized helper data to decrypt a watermarked version of the content that was sent to him. Still, the client cannot extract the watermark out of the helper data or obtain the original content in the clear.

In this paper, we show how secure watermark embedding can be realized by utilizing concepts of partial encryption [12], which have primarily been developed

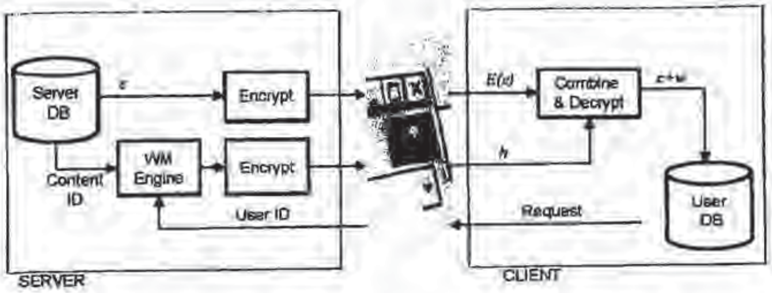


**Fig. 1.** Electronic Content Distribution utilizing secure watermark embedding.

in the past in order to speed up the encryption process of media files by selectively encrypting only the perceptually most relevant parts. To use partial encryption for secure watermark embedding, we encrypt the perceptually most relevant parts of a piece of content and give the client helper data which allows him to decrypt the content in a slightly different way; the differences induced by the changed decryption process represent the watermark. In this paper, we show how this general methodology can be applied to baseband audio and MPEG-2 compressed video streams.

The rest of the paper is organized as follows. In Section 2 we discuss in greater detail the concept of and the requirements for practical secure watermark embedding; Section 3 reviews existing client-side watermark embedding solutions with regard to the requirements. In Section 4 we outline our general methodology for secure embedding, while Sections 5 and 6 discuss two concrete implementations of the methodology for baseband audio and MPEG-2 encoded video streams. Finally, Section 7 concludes the work.

## 2 Secure Client-Side Watermark Embedding

Figure 1 illustrates the concept of secure watermark embedding in the context of electronic content distribution in greater detail. When a client wants to retrieve a piece of content $c$, he contacts a distribution server, who ships an encrypted version $E(c)$. At a later state, some party (not necessarily the same server) generates a watermark representing the identity of the user and computes helper information $h$, implicitly coding the personalized watermark. This helper information is subsequently shipped to the client, who can use $h$ to decrypt a copy of the content which is watermarked by $w$ (denoted by $c+w$ in the figure); however, the helper information $h$ does not allow him to infer either $c$ or the watermark directly.

We can identify the following requirements for practical secure watermark embedding techniques:

- *Low bandwidth overhead.* The transmission overhead induced by the secure watermark embedding mechanism should be as small as possible. In particular:
    - The employed encryption algorithm should operate in a space efficient manner, i.e., the size of $E(c)$ should be similar to the size of $c$. This is especially relevant as content is usually transmitted in (lossy) compressed form. The chosen encryption algorithm $E$ should thus ideally operate directly on compressed content.
    - The bandwidth required for the transmission of the helper data $h$ should be considerably smaller than the one required for transmitting $E(c)$.
- *Security.* Transmitting $E(c)$ and the helper data $h$ must not compromise the security of either $c$ or $w$. In particular, $h$ must not reveal to the client more information about the original and the watermark than it is already leaked by the watermarked work itself.
- *Content independence.* Ideally, $h$ should be independent of the content $c$. This enables the use of secure watermark embedding in flexible distribution models that split the content distribution from the license acquisition process. Furthermore, it allows to pre-compute helper data for a particular set of clients (which may allow to implement live video broadcasting solutions in which the computationally intensive process of helper data generation can be done offline).

## 3 Related Work

Secure watermark embedding has only recently gained attention in the scientific community. With current technology, client-side watermark embedding is typically performed in a dedicated piece of hardware within consumer electronic devices (see [11, 10] for a framework). However, this solution has the apparent drawback that it requires a dedicated hardware installed base, cannot be easily integrated in legacy applications and is not easily updatable. Thus software solutions are clearly preferable.

In broadcast environments, Crowcroft et al. [4] and Parviainen et al. [9] proposed a client-side watermark insertion technique based on stream switching. In their method, they chop the content stream into small chunks and broadcast two version of the stream, watermarked with different watermarks. Each chunk is encrypted by a different key. Clients are given a different set of decryption keys that allow them to selectively decrypt chunks of the two broadcast streams such that each client obtains the full stream. The way the full stream is composed out of the two broadcast versions encodes the watermark. However, this solution does not meet the bandwidth requirements stated above, as the amount of data needed to be broadcast to the clients is twice as large as the content itself.

Emmanuel et. al. [5] proposed a client-side embedding method in which a pseudorandom mask is blended over each frame of a video; each client is given a different mask, which, when subtracted from the masked broadcast video, leaves an additive watermark in the content. The scheme has security problems, as

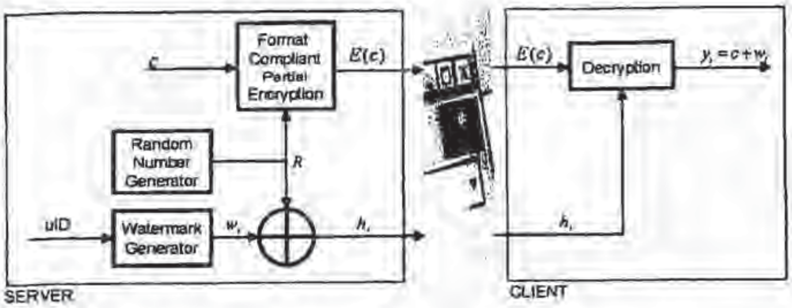


**Fig. 2.** Secure watermark embedding using partial encryption.

a constant mask is used for all frames of a video, which can be estimated by averaging attacks. Subsequently the estimated mask can be subtracted from the encrypted video in order to obtain a perceptually acceptable and watermark-free version of the content.

Anderson et. al. [3] designed a special stream cipher, called Chameleon, which allows, by appropriate design of encryption keys, to decrypt Chameleon-encrypted content in slightly different ways. Thus, the special design of the cipher allows to leave a key-dependent trace in the decrypted data stream. Kundur and Karthik [6] were the first to use techniques from partial encryption together with Chameleon in order to fingerprint digital images. Their method is based on encrypting the signs of DCT coefficients in an image; during decryption some signs are left unchanged, which leaves a detectable fingerprint in the image. As the sign bits of DCT coefficients are perceptually significant, the partially encrypted version of the content is heavily distorted. However, as some DCT coefficients are left scrambled during decryption, the watermark can be visible; visibility of the watermark must be traded in for optimal detection.

Recent work by Adelsbach et. al. [1] showed how to generalize the Chameleon cipher in order to be able to embed spread spectrum watermarks. However, the work still only considers uncompressed baseband signals.

## 4 Secure Embedding Through Partial Encryption

In this section, we show how secure watermark embedding can be realized through partial encryption. As mentioned above, we choose a partial encryption scheme and encrypt perceptually important parts of the content, while preserving the content file format. Finally, we provide the client with helper data, which allows him to access a personalized, slightly modified version of the content. The remaining unique signature (difference between the original and the reconstructed version) can later be used as a forensic watermark to trace back the origin of the content. The concept is schematically depicted in Figure 2.

Note that in our approach we only perform partial encryption of the content $c$ (for example, as opposed to [1]). Typically, in DRM applications partial encryption of the content is sufficient, as the content itself is not confidential (it can be accessed by every legitimate user). For the security analysis of a forensic tracking watermarking architecture one has to assume that an attacker posesses at least the same information as a legitimate user. Thus, the applied encryption scheme only needs to protect those parts of the content that potentially help an attacker to derive an un-watermarked copy. In addition, partial encryption has the advantage that the encrypted files can be viewed or listened on a normal playback device. Even though the content is severely distorted, the user gets a first impression on how the decrypted content will look like. Thus, the partially encrypted content can serve as a low-quality preview.

In greater detail, the proposed system works as follows:

- **Server**: The server performs the following operations:
  1. The server reads an input content $c$,
  2. chooses perceptually significant features of $c$,
  3. and encrypts those features using a format compliant partial encryption scheme; this process yields to a perceptually unacceptable distorted content $E(c)$, which can be safely released into the public. The features are chosen in such a way that it is hard to reconstruct, using techniques of signal processing, a perceptually acceptable estimate of $c$ out of the encryption $E(c)$.
  4. For each user $i$, the server generates a watermark $w_i$ and chooses helper information $h_i$, which can be applied to $E(c)$ in order to undo the distortions of the encryption process and to leave a detectable watermark $w_i$. The helper information $h_i$ is constructed in such a way that knowledge of $h_i$ does not allow the client to infer the watermark. In addition, knowledge of $h_i$ does not facilitate obtaining an un-watermarked copy of the content.
  5. Finally, the server sends $h_i$ to the client.
- **Client**: The client performs the following operations:
  1. The client acquires the content $E(c)$ from the public domain and
  2. receives the helper information $h_i$ from the server via a one-to-one link.
  3. Finally, the client applies $h_i$ to the distorted content $E(c)$ in order to obtain his personalized copy of the content $y_i$. This process produces a perceptually acceptable, but watermarked output signal, $y_i = h_i(E(c)) = c + w_i$.

In the following sections, we show how this general concept can be applied to baseband audio and MPEG-2 encoded video streams by discussing two proof-of-concept implementations.

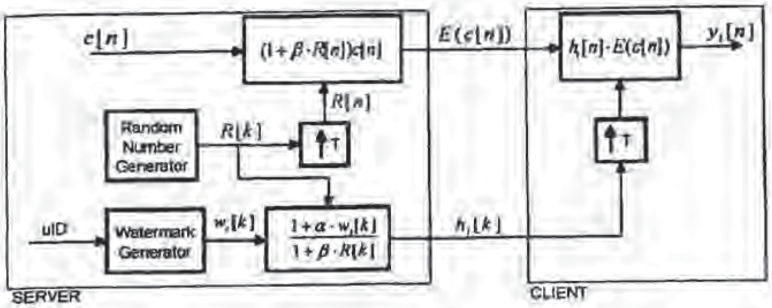


Fig. 3. MASK watermarking system based secure watermarking scheme.

## 5 Baseband audio

In this section, we show how the MASK [7] audio watermark embedder can be implemented safely at an untrusted client device. To facilitate our discussion, we first present a brief summary of the MASK watermarking system, and subsequently show how this system is implemented in the context of secure watermark embedding.

*The MASK watermarking system.* In MASK, a watermark is embedded by modifying the envelope of the host signal. More specifically, given the host signal $c[n]$ and the watermark signal $w_i[n]$, the watermarked content $y_i[n]$ is given by

$$y_i[n] = c[n] + \alpha[n] w_i[n] c[n], \tag{1}$$

where the watermark signal $w_i[n]$ is chosen in such a way that it predominantly modifies the short time envelope of the signal, and the gain function $\alpha[n]$ is controlled by a psychoacoustic model of the human auditory system. The MASK system has been extensively tested and has proven to combine good audibility quality with high robustness. For more details on the implementation and on the robustness tests, we refer to [7].

*Joint decryption and watermarking.* Figure 3 shows the secure embedding framework for MASK. Encryption of the original content is achieved by modulating the host signal with a piece wise stationary random sequence $R[k]$ such that the resulting audio is perceptually annoying to listen to. Let $T$ be the interval (frame) over which $R[k]$ remains constant and let $c_k[n]$, $0 \leq n \leq T-1$, represent the $k$-th frame of the content signal. We encrypt the $k$-th frame by

$$E(c_k[n]) = (1 + \beta[k] R[k]) c_k[n], \tag{2}$$

where the weighting coefficient $\beta[k]$ is chosen in such a way that the condition $1 + \beta[k] R[k] \neq 0$ is always satisfied.

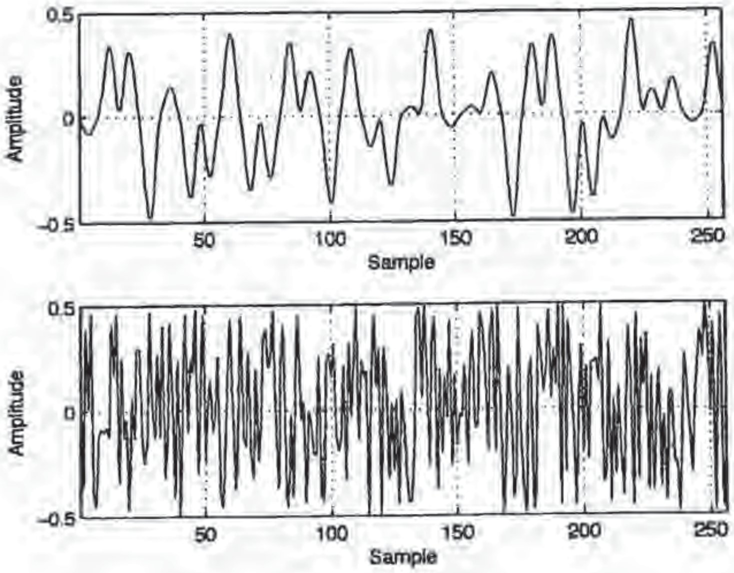


**Fig. 4.** Typical wave shapes of $w_i[k]$ (top) and $R[k]$ (bottom).

For one client $i$, the server first generates the MASK watermark signal $w_i[k]$ that is linked to the identity of the client (for the process of payload encoding we refer to [7]). The watermark signal $w_i[k]$ is made to vary gracefully in order to minimize audible artifacts in the watermarked content. The typical waveform of $w_i[k]$ is shown in the upper part of Figure 4. Finally the server computes a helper signal $h_i$ for user $i$, which is given by

$$h_i[k] = \frac{1+\alpha[k]w_i[k]}{1+\beta[k]R[k]}, \tag{3}$$

and distributes this signal to client $i$.

On the client side, joint decryption and watermarking is achieved by taking the product between the helper data $h_i[k]$ and the encrypted frame content $E(c_k)$. More specifically, for each frame $k$, the client computes the watermarked frame signal $y_{i,k}[n]$ by

$$y_{i,k}[n] = h_i[k]E(c_k[n]). \tag{4}$$

Substituting the values of $h_i[k]$ and $E(c_k[n])$ from (3) and (2), respectively, we obtain

$$y_{i,k}[n] = (1+\alpha[n]w_i[k])c_k[n]. \tag{5}$$

From the last equation we see that the client is left with a MASK-watermarked version of the content. The MASK watermarking system is extensively studied in different papers [8, 7, 14] and has been shown to combine excellent audibility/robustness tradeoff. Thus, in this paper, we do not consider such details, interested readers are advised to visit the above references.

*Effect of the spreading factor on Robustness and Security.* Note that in the above, we have assumed that the random number $R[k]$ remains constant for a period of $T$ samples. If we let $q$ represent the number of audio channels, this means that a single random number is provided for every $T \times q$ audio samples. This in turn implies that the size overhead introduced by the helper data is linearly related to the "spreading" factor $T$. In MASK system (crf. [7]), $T$ represents the so-called watermark symbol period. It reflects the granularity of the watermark symbol repetition. If the audio clip is long-enough the symbol period does not affect the robustness significantly because the total number of samples per a single watermark symbol remains unchanged. To be more specific, let $T_1$ and $T_2$ be two spreading factors, $L_w$ be the length of the watermark sequence and $L_s \gg L_w \times \max(T_2, T_1)$ be the length of the audio clip. Then, in the audio, the watermark sequence will be repeated $r_1 = L_s/(L_w * T_1)$ times for the case of $T_1$ and $r_2 = L_s/(L_w * T_2)$ times for the case of $T_2$. The repetition of each watermark symbol is given by $T_1 \times r_1$ for the first case and by $T_2 \times r_2$ for the second case. After substituting the values of $r_1$ and $r_2$, both of the above products simplify to $L_s/L_w$. This shows that if $L_s$ is large enough, the level of averaging used to extract each symbol is independent of the spreading factor and thus robustness is not significantly affected. However, the spreading factor $T$ introduces tradeoff between security and size overhead. That is, repeating $R[k]$ over several samples leaks information. We differ the security analysis for a future work.

*Experimental results.* We have tested the system depicted in Figure 3 using different stereo audio streams sampled at 44.1 kHz. For the test, we have chosen $T = 64$ samples, $\beta[k] = \beta = 0.9$ and $\alpha[k] = \alpha = 0.15$. The encrypted audio $E(c)$, though still recognizable, is graded as extremely annoying to listen to, whereas the watermarked output signal $y_i$ is perceptually indistinguishable from the original one. In the implementation, the helper data was coded in 8 bits float, thus for the transmission of the helper data a side channel with capacity of at least

$$C_{CH} = \frac{8 * 44100}{T} \text{ bps}$$

is required. For $T = 64$, this equals to 5.5 kbps. Compared to a bitrate of a typical compressed audio stream (about 128 kbps), this amounts to an overhead of approximately 6%.

## 6 MPEG-2 compressed video

In this section, we show how the general methodology of joint watermarking and decryption can be applied to MPEG-2 compressed streams. Again, we first describe the employed watermarking scheme and subsequently detail how it is used in conjunction with a partial encryption scheme.

*Watermarking scheme.* We use an additive spread spectrum watermark which modulates the luminance DC values of all I-frames present in the MPEG-2 stream. Recall that in MPEG-2, each frame is divided into $N \times M$ macroblocks, each having $16 \times 16$ pixels; a macroblock is further subdivided into four $8 \times 8$ luminance blocks. Let $c_k[x, y]$, $1 \leq x \leq 2N$ and $1 \leq y \leq 2M$, denote the luminance DC values of all image blocks of the $k$-th I-frame. As a carrier for the watermark, a pseudorandom bit pattern of size $N \times M$, where each value is either $+1$ or $-1$, is created. To encode a payload, the pattern is shifted circularly both in the horizontal and the vertical direction to obtain a watermark $w_i$ of size $N \times M$. From $w_i$, we obtain a $2N \times 2M$ matrix $w_i'$ by

$$w_i' = w_i \otimes \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix},$$

where $\otimes$ denotes the Kronecker product. The watermark $w_i'$ is used to modulate the luminance values $c_k[x, y]$ to obtain the watermarked content

$$y_k[x, y] = c_k[x, y] + \alpha w_i'[x, y],$$

where $\alpha$ controls the watermark embedding strength. This embedding method has the effect that the upper two DC values in a macroblock will be modulated with the watermark, whereas the lower two values are left unchanged (and will be used in the detection process to minimize the influence of the host signal on the watermark detection result).

For watermark detection, the stream is decompressed and a constant number of consecutive frames is averaged; a blockwise DCT transform is applied to this averaged frame. In each macroblock, the upper two (watermarked) luminance DC coefficients are added, from which the lower two (unchanged) coefficients are subtracted. This way, the averaged frame is condensed to an $N \times M$ matrix, which is finally correlated with circular shifts of the watermark pattern $w_i$. If sufficient correlation exists, the watermark is assumed to be present; the shift with which the highest correlation has been achieved codes the payload. Note that for simplicity of explanation, we have used a constant watermark for all I-frames. However, the system can be easily changed to support embedding of different watermarks in subsequent I-frames.

*Joint decryption and watermarking.* To encrypt an MPEG-2 stream, we produce for each I-frame a random $2N \times 2M$ matrix $r_{i,k}$ with entries in the range of $(-2^l, 2^l)$ and add its elements to the luminance DC coefficients

$$E(c_k[x, y]) = c_k[x, y] + r_{i,k}[x, y].$$

Depending on the value of $l$, this results in more or less severe visible artifacts in the stream; the visual effect of this partial encryption method is illustrated in Figure 5. Part (a) of the figure shows a frame of the video, while (b) illustrates the effect of the chosen partial encryption: due to the noise in DC values, severe blocking artifacts are introduced.

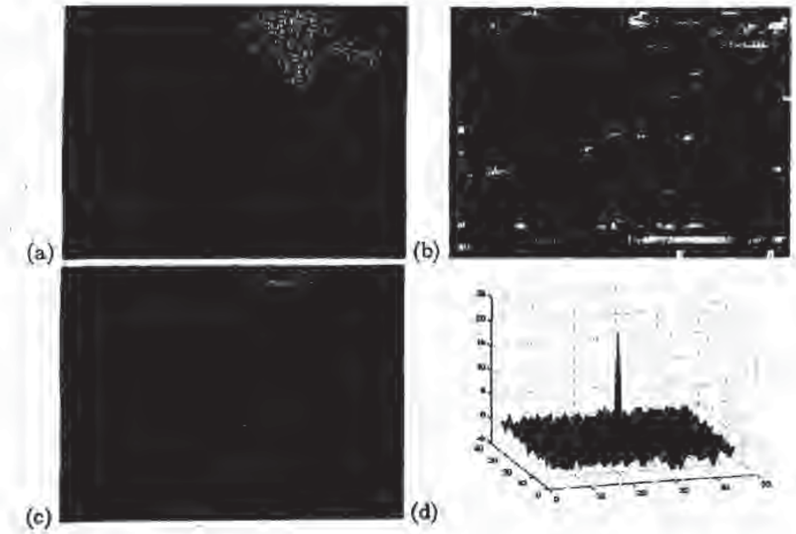


**Fig. 5.** Illustration of the proposed combined watermarking and decryption system: (a) an original frame of a MPEG-2 compressed movie, (b) the corresponding encrypted frame, (c) the reconstructed watermarked frame and (d) the watermark detection result.

For secure watermark embedding, the client is given the encrypted version of the stream as well as (for each I-frame) the $2N \times 2M$ matrix $h_{i,k} = r_{i,k} - \alpha w'_i$ as helper information, which he subtracts from the DC luminance coefficients to obtain the watermarked content:

$$\begin{aligned} y_k[x,y] &= E(c_k[x,y]) - h_{i,k}[x,y] \\ &= c_k[x,y] + \alpha w'_i[x,y]. \end{aligned} \tag{6}$$

Figure 5(c) shows that using equation (6), the visual artifacts can be completely removed in the joint decryption and watermarking step. Still, the watermark can be reliably detected by a correlation detector, see part (d) of the figure.

*Experimental results.* We have tested the system on several MPEG-2 compressed movies; results for four different clips are summarized in Table 1. First, we can note that embedding the watermark only marginally increases the size of the compressed content (about 0.05%). The encryption step has a noticeably effect on the size of the content, as it is adding uniformly distributed noise. Depending on the strength of the noise (i.e., the value $l$) we can observe an increase in the content size of about $0.5 - 0.8\%$. The size of the helper data which needs to be sent to the client in addition to the content scales linearly with the content

| clip | original size (bytes) | watermark overhead | overhead for $l = 3$ | | overhead for $l = 3.5$ | |
|---|---|---|---|---|---|---|
| | | | encryption | helper data | encryption | helper data |
| A | 13,561,344 | 0.05% | 0.53% | 1.11% | 0.77% | 1.30% |
| B | 14,998,551 | 0.03% | 0.45% | 1.10% | 0.70% | 1.29% |
| C | 12,808,526 | 0.03% | 0.48% | 1.10% | 0.73% | 1.28% |
| D | 15,007,249 | 0.02% | 0.25% | 1.12% | 0.45% | 1.30% |

**Table 1.** Performance of the combined watermarking and decryption system.

size: for each luminance DC value of the content, one $l$-bit value needs to be transmitted. For $l = 3$, this amounts to a helper data size of about 1.1% of the content, whereas for $l = 3.5$, we obtain an overhead of about 1.3%.

## 7 Conclusions and Future Work

In this paper, we considered secure watermark embedding algorithms, which allow to securely insert a watermark at an untrusted client device without compromising the security of the watermark key, the watermark or the original content. To implement the functionality, we perform a partial encryption of the content and give the client helper information, which allows to decrypt a slightly different version of the content; the differences between the original and the reconstructed version constitute a forensic watermark. In particular, we discussed two proof-of-concept implementations, one for the MASK watermarking scheme operating on baseband audio and one for a simple additive spread spectrum watermark operating on MPEG-2 compressed video streams. We showed that partial encryption can overcome the major current obstacle of secure watermark embedding, namely limit the size of the helper data needed to be transmitted between the server and the client. In the current paper, we have mainly concentrated on efficiency aspects of secure watermark embedding and have not thoroughly addressed security issues of the employed partial encryption (i.e., the exact relation between the difficulty of a successful cryptanalysis and the complexity of watermark removal). We leave this, as well as the investigation of different partial encryption methods, for future work.

## References

1. A. Adelsbach, U. Huber, and A.-R. Sadeghi. Fingercasting—joint fingerprinting and decryption of broadcast messages. In *11th Australasian Conference on Information Security and Privacy*, 2006.
2. Open Mobile Allowance. OMA digital rights management. http://www.openmobilealliance.org.
3. R. J. Anderson and C. Manifavas. Chameleon—a new kind of stream cipher. In *FSE '97: Proc. of the 4th Int. Workshop on Fast Software Encryption*, pages 107–113, London, UK, 1997. Springer-Verlag.

4. J. Crowcroft, C. Perkins, and I. Brown. A method and apparatus for generating multiple watermarked copies of an information signal. WO Patent No. 00/56059, 2000.
5. S. Emmanuel and M.S. Kankanhalli. Copyright protection for MPEG-2 compressed broadcast video. In *ICME 2001. IEEE Int. Conf. on Multimedia and Expo.*, pages 206–209, 2001.
6. D. Kundur. Video fingerprinting and encryption principles for digital rights management. *Proceedings of the IEEE*, 92(6):918–932, 2004.
7. A.N. Lemma, J. Aprea, W. Oomen, and L. van de Kerkhof. A temporal domain audio watermarking technique. *IEEE Transactions on Signal Processing*, 51(4):1088–1097, 2003.
8. Aweke Negash Lemma, Javier Aprea, Werner Oomen, and Leon v.d. Kerkhof. A robustness and audibility analysis of a temporal envelope modulating audio watermark. In *IEEE DSP/SPE workshop proceedings*, Gallaway Gardans, GA, USA, October 13-16 2002.
9. R. Parviainen and P. Parnes. Large scale distributed watermarking of multicast media through encryption. In *Proceedings of the International Federation for Information Processing, Communications and Multimedia Security Joint working conference IFIP TC6 and TC11*, pages 149–158, 2001.
10. P. Tomsich and S. Katzenbeisser. Copyright protection protocols for multimedia distribution based on trusted hardware. In *Protocols for Multimedia Systems (PROMS 2000)*, pages 249–256, 2000.
11. P. Tomsich and S. Katzenbeisser. Towards a robust and de-centralized digital watermarking infrastructure for the protection of intellectual property. In *Electronic Commerce and Web Technologies, Proceedings (ECWEB 2000)*, volume 1875 of *Springer Lecture Notes in Computer Science*, pages 38–47, 2000.
12. A. Uhl and A. Pommer. *Image and Video Encryption, From Digital Rights Management to Secured Personal Communication.* Springer, 2005.
13. M. van der Veen, A. Lemma, and A.A.C. Kalker. Electronic content delivery and forensic tracking. *Multimedia Systems*, 11(2):174–184, 2005.
14. Michiel van der Veen, Aweke Lemma, and Ton Kalker. Watermarking and fingerprinting for electronic music delivery. In *SPIE Workshop 2004*, San Jose, CA, USA, 2004.

# SELF-PROTECTING DIGITAL CONTENT

— A TECHNICAL REPORT FROM THE CRI CONTENT SECURITY RESEARCH INITIATIVE —

Paul Kocher, Joshua Jaffe, Benjamin Jun, Carter Laren, Nate Lawson

Keywords: Piracy, risk management, watermarking, renewability, programmable security, forensic marking

---

## EXECUTIVE SUMMARY

### Introduction

Despite the high public profile of piracy as a threat to intellectual property owners, surprisingly little useful research has been done to understand the range of technical solutions that are feasible. This paper presents results from a study sponsored by Cryptography Research, Inc. to determine how cryptographic systems can provide the most effective long-term deterrent to the piracy of digital video and other content distributed on optical media.

Although numerous products and technologies have been advertised as solutions to the problem of piracy, most commercial security systems fail catastrophically once an implementation is compromised. These designs can work in limited deployments, but any technology deployed as part of a major standard will inevitably attract extremely determined attacks – and some implementations will get broken. The long lifespan of media formats, diversity of player implementations, complexity of security/usage models, and constantly-changing risk scenarios provide attackers with numerous avenues of attack and the time and resources to explore them. As a result, effective content protection systems must be able to survive compromises and adapt to new threats.

### Risk Managing an "Unsolvable Problem"

Risk management approaches often provide the only way to manage security problems in situations where unbreakable solutions are unavailable or impractical. For example, the major credit card networks are based on fundamentally insecure magnetic stripe technology, yet risk management efforts have held fraud rates below 0.1 percent. Similarly, computer security flaws are discovered frequently, but users can manage (though not eliminate) their risk by applying software updates and by using anti-virus programs. Without risk management tools, neither credit/debit networks nor the Internet could survive.

Piracy, like credit card fraud and computer security, is a problem that cannot be solved completely. Our research identified technical systems that give content owners the ability to control their risk. The most practical and effective of these combine programmable code with encrypted digital content. This code would be distributed as part of the content, execute dynamically during playback, and enforce each title's security policies. Publishers could then control security for their own content.

### Programmable Security: Smart Content

Programmable security approaches give publishers the freedom to add new countermeasures and improve security after a standard has been widely

**Examples of correctable problems with existing content protection systems:**

- After players are sold, security is static and cannot evolve as new attacks and new threats appear.
- Compromises beyond the decoder (digital output devices, software device drivers, etc.) are not recoverable.
- Product vendors do not receive clear benefits for investing in security.
- Copies cannot be traced to decoders for revoking equipment, reducing pirates' anonymity, or helping with prosecution.

adopted. Players would include a simple virtual machine with APIs that provide data about the playback environment, such as player information, software versions, output device types, and user commands. The content-specific code would analyze this data and control whether and how decoding would proceed. The code can also use player APIs to authenticate output devices, support player-specific security features, validate user actions (e.g., copy vs. play), check whether media is consumer-recordable, and implement locale-specific requirements. Content being decoded by software-based PC players could even check for malicious software or device drivers. Playback can be prevented if the environment is unacceptable.

**The Chess Game: Avoiding Checkmate**

The security flaws in the system used to protect DVD video cannot be fixed without abandoning compatibility with the installed base of DVD players. Programmable protection systems have a unique ability to avoid this category of problem by shifting responsibility for security from players to the content itself. While compromises will still occur, new titles can carry security code that corrects for past vulnerabilities. As a result, each attack has an effective response. Content owners will be able to constantly upgrade security over time – even to correct for risks that were not known when the original system was designed.

Although risk management can control problems, no security technology can eliminate piracy. Some attacks, such as copying from analog outputs (speakers, displays, etc.) using general-purpose recording devices, are impossible to prevent completely and will always remain a threat. Similarly, no player or media technology can eliminate piracy using Internet-based file sharing networks. When problems do occur, self-protecting content can be used to correct security weaknesses and to identify/revoke pirates' equipment, although responses to the most determined pirates will continue to require law enforcement.

**Economics of Security**

Today, product manufacturers generally bear the costs of providing security, but do not receive the benefits. As a result, vendors lack incentives for making significant investments in controlling piracy. Placing security code on the media helps correct this economic imbalance by giving content owners responsibility for the security software used by their own content. This also gives manufacturers incentives to become active participants in security because only well-designed players will be trusted by publishers with their most compelling content. Publishers can use their control over each title's security to manage their risk and maximize profits.

**Forensic Marking: Uncovering Pirates**

Effective risk management requires the ability to detect and to respond to problems. While media-based security code makes it possible for new content to resist known attacks, publishers must also be able to gather information from past compromises. Watermarks have been proposed for carrying security-related information. Unfortunately, it appears to be infeasible to make a watermark that is secure against removal by adversaries who have reverse engineered the mark detector. More generally, we do not believe that conventional ("public") watermarks will prove effective as a robust way to block copying in widely-deployed standards.

Fortunately, a new class of steganographic marks provides an attractive alternative to conventional watermarks for risk management purposes. These "forensic marks" are embedded dynamically and can carry detailed information about the decoding process. Unlike conventional watermarks, forensic marks can be provably secure, efficient to embed, imperceptible, and extremely robust.

Publishers can analyze mark contents to determine the specific equipment and methods used to make each pirated copy. This data is essential for rights holders to be able to revoke devices used for piracy, improve the security of future content, and prosecute pirates. Because forensic marks embed identifying information in decoded (analog) output, they have the

## Table of Contents

psychological benefit of reducing the perceived anonymity and safety of piracy without affecting the privacy of legitimate users.

## Need for Leadership

Investments in security have been inadequate relative to the major economic threat posed by piracy. After successfully lobbying for the Digital Millennium Copyright Act, publishers have failed to present a coherent long-term technical strategy.

Efforts to improve security will require strong technical leadership. Without clear objectives, standards efforts tend to degenerate into unwieldy and ineffective committees with short-term focus. Leadership is also needed to verify that security needs are met before products ship and to help secure designs succeed in the marketplace. We conclude that only rights holders can provide this leadership; no other participants have the motivation, expertise, or resources to ensure the deployment of effective anti-piracy technologies.

* * *

**Cryptography Research, Inc.** provides consulting services and technology to solve complex security problems. In addition to security evaluation and applied engineering work, CRI is actively involved in long-term research in areas including tamper resistance, content protection, network security, and financial services. This year, security systems designed by Cryptography Research engineers will protect more than $50 billion of commerce for wireless, telecommunications, financial, digital television, and Internet industries. For additional information or to arrange a consultation with a member of our technical staff, please contact Jennifer Craft at 415-397-0329 or visit www.cryptography.com.

**Paul Kocher** is President and Chief Scientist of Cryptography Research. His work includes designing numerous cryptographic applications and protocols, including SSL v3.0, the world's most widely used security protocol. In addition to leading the team at CRI that discovered differential power analysis and designed the record-breaking DES key search machine "Deep Crack", he is also credited with discovering timing attack cryptanalysis and co-founding ValiCert, Inc. (NASDAQ:VLCT). His work has been reported in forums ranging from technical journals and Scientific American to CNN and the front page of the New York Times. Paul can be contacted via e-mail at paul@cryptography.com.

**Josh Jaffe** is a Security Architect at Cryptography Research, Inc. who specializes in signal processing applications, cryptographic implementations. He holds B.S. degrees in computer science and physics/astronomy from Brandeis University. Josh can be reached via e-mail at josh@cryptography.com or at 415-397-0324.

**Benjamin Jun** is Vice President of Cryptography Research and is responsible for consulting services and content protection efforts. He has developed and evaluated numerous systems for the protection of financial transactions, audio content, and pay television. Prior to Cryptography Research, Ben worked at IDEO Product Development on Secure Content Distribution Systems. He has also held positions at Bain & Company, the National Institute of Standards and Technology, and the Institute for Defense Analysis. Ben holds B.S. and M.S. degrees in Electrical Engineering from Stanford University, where he is an NSF Graduate Fellow and a Mayfield Fellow. Ben can be contacted at (415) 397-0323 or at ben@cryptography.com.

**Carter Laren** is a System Architect at CRI with a background in electrical engineering and extensive experience designing and implementing hardware and software cryptographic components. Prior to joining Cryptography Research, Carter worked at L-3 Communications where he designed secure communication systems for both government and commercial applications. He also held the position of Weapon Systems Engineer at Lockheed Martin, where he designed and tested portions of the AEGIS Combat System. Carter is a Chancellor's Scholar at the University of Pittsburgh, where he received a B.S. degree in Electrical Engineering. Carter can be contacted at (415) 957-2667 or at carter@cryptography.com.

**Nate Lawson** is a Senior Security Engineer at Cryptography Research, Inc. with a background in systems engineering and network security. Prior to joining Cryptography Research, Nate designed and implemented network devices, intrusion detection systems, SAN appliances, and media distribution networks for companies including ISS, Decru, and Nifty Devices. He also co-founded Elite Networking (http://elite.net). Nate can be contacted at (415) 397-8662 or nate@cryptography.com.

**The Content Security Research Initiative** is an ongoing effort funded by Cryptography Research, Inc. to solve security problems for the content distribution industry. This effort has yielded significant advances in securing pay television broadcasts, Internet downloads, and optical media. Results from the study (including approaches in this paper) are protected by U.S. patents #6,298,442, #6,327,661, #6,304,658, #6,188,766, #6,289,455, #6,381,699, and/or #6,278,783; other U.S. and international patents are pending, including U.S. patent application 20020141582 and U.S. provisional application 60/279,323 (which specifically cover programmable self-protecting content technologies). Please contact Cryptography Research for more information about the initiative, other research results, or technology/patent licensing.

## 1. INTRODUCTION

If hard drive densities continue to double annually, a drive costing $250 in 2012 will be able to store 160 terabytes – enough for over 10,000 full-length high-definition movies plus 100,000 uncompressed CDs.[1] Similar improvements in communication technology will provide users with the bandwidth required to utilize this storage capacity. These advances are presenting increasingly complex risks and challenges for those wishing to limit piracy and profit from their intellectual property.

Some have argued that the pirates will prevail, because all content will eventually be available as "unprotected bits" that can be copied easily and anonymously. For example, one cryptographer has argued that, "All digital copy protection schemes can be broken, and once they are, the breaks will be distributed... Average users will be able to download these tools from Web sites that the laws have no jurisdiction over."[2]

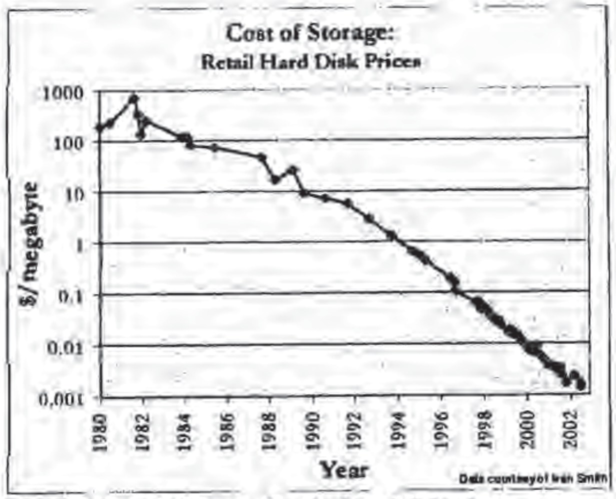
Our research challenges these dire predictions and examines the question of how security technologies can most effectively control piracy in the long-term while satisfying the needs of consumers and device manufacturers. Although our results support the view that the total elimination of piracy is not a realistic objective, we believe that properly-designed technical systems can provide an effective deterrent and prevent piracy from destroying the value of digital content.

Cryptography developed from the need to keep information private. In many ways the field is very advanced – the best modern cryptographic algorithms are flexible, efficient, reliable, and virtually unbreakable. Even an attacker with the entire world's computing power, access to virtually unlimited amounts of encrypted data, and the best known attack methods cannot break a single strongly-encrypted message.

Strong algorithms do not necessarily make systems secure. Weaknesses in the protocols and products that manage keys and decrypted content make it unnecessary for attackers to break the underlying cryptographic algorithms. Unfortunately for content distribution systems, implementation weaknesses are so common that compromises are virtually inevitable.

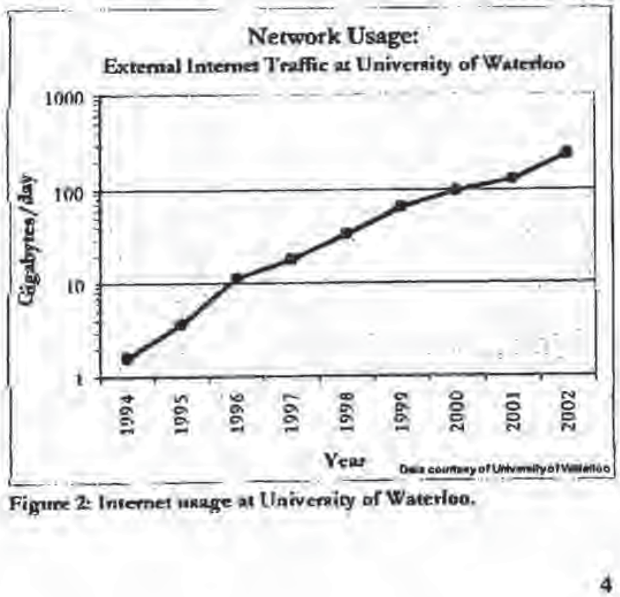
The primary technical challenge is therefore to design architectures that maintain their effectiveness even after individual devices or implementations have been compromised. Protection measures that fail catastrophically when attacked are clearly not acceptable as long term solutions. In contrast, even relatively easy-to-break approaches may be useful if they provide a lasting deterrent to low-budget or casual piracy and limit the problem to professional operations that can be targeted by investigative and legal efforts.

This paper presents results from a study sponsored by Cryptography Research, Inc. to determine



Data courtesy of Ivan Smith

Figure 1: Cost of storage – advertised hard disk prices.



Data courtesy of University of Waterloo

Figure 2: Internet usage at University of Waterloo.

---

[1] (10,000 movies × 9 gigabytes) + (100,000 CDs × 650 megabytes) = 155 terabytes. A 160 gigabyte drive cost $250 in July 2002. A similarly-priced drive in 2012 is expected to hold 160 terabytes.

[2] Schneier, Bruce, "The Futility of Digital Copy Prevention," Cryptogram, May 15, 2001.

whether technical systems can provide a meaningful long-term deterrent to piracy. The examples in this paper focus primarily on the problem of securing video distributed on conventional (passive) optical media, although our results are also applicable to broadcast/Internet distribution and other content types. We do not address philosophical questions such as whether artists should be able to apply copy protection to their work.

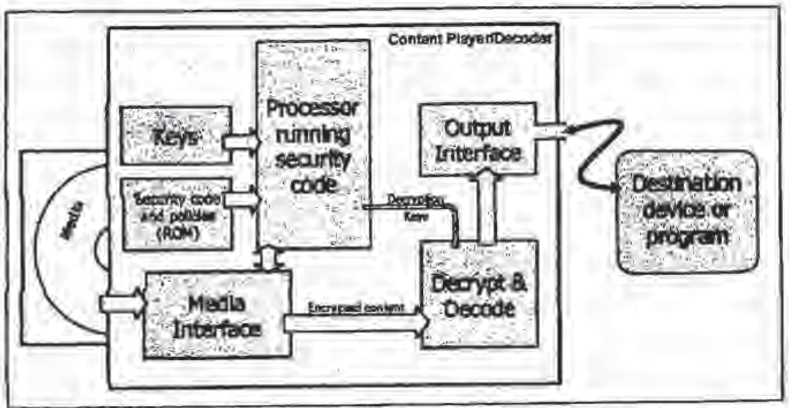


Figure 3: Architecture of a conventional content player.

## 2. CSS & OTHER CONVENTIONAL ARCHITECTURES

The Content Scramble System[3] (CSS) used for DVD video is noteworthy because of its widespread use and poor design. CSS is implemented in the player and provides a simple, fixed security policy for all content: any device with valid keys can decrypt all media valid in its region.

Figure 3 shows the architecture of a typical player implementing a conventional content encryption scheme such as CSS. The content is compressed, encrypted, then distributed on read-only media. To allow off-line playback, every player is pre-loaded with all keys required to decrypt all media it will ever decode. The security scheme is defined in the player, typically as software, and enforces a set of fixed security rules. After decryption, the content is sent to an output interface, which is typically unprotected or has protection features that are independent of the protection used on the media.

CSS failed to meet even its limited security objectives. Although CSS contains many design flaws, the most catastrophic was the use of proprietary cryptographic algorithms which proved trivial to break. After a player compromise, CSS was supposed to allow new DVDs to be mastered so that they could not be decoded by players with revoked manufacturer keys. Poor use of cryptography allowed attackers to circumvent this capability. Today, circumvention software is widely available, but CSS cannot be repaired without making the entire installed base of DVD players obsolete. In practice, CSS would probably have failed even without the obvious cryptographic weaknesses, as consumers would not have tolerated the revocation of a major manufacturer. Other limitations of CSS include its inability to revoke individual decoders, adapt security policies to new threats, secure/revoke digital output formats, or trace pirated content back to a compromised device.

The security problems in CSS can be traced back to the design process. CSS was developed by product companies without major exposure to piracy or adequate experience designing secure systems. The Copy Protection Technical Working Group (CPTWG), which was supposed to ensure the security of DVD, was politically divided and lacked leadership or active participation by experienced cryptographers or security engineers. As a result, the CSS specification failed to provide adequate assurance of its own security, yet unrealistically assumed bug-free implementations.

Because CSS failed to give implementers clear incentives to ensure security, implementation quality became an increasingly major problem after the success of the DVD format was assured.[4] Some vendors even appear to have intentionally produced insecure products to help users circumvent the CSS region coding. For example, the region coding on many players can be defeated by pressing a "secret" sequence of buttons.[5] The source of the problem is that manufacturers profit from sales to people who circumvent the region coding, but do not incur losses when their products are broken.

---

[3] The official specifications for CSS (also called Content Scrambling System) are confidential and are licensed by the DVD Copy Control Association (http://www.dvdcca.org).

[4] Cryptography Research ultimately discontinued auditing CSS implementations because vendors wanted documentation that their products were not the "least secure" on the market, and were not interested in identifying and correcting security problems.

[5] Numerous web sites specialize in documenting these sequences. See, for example, http://www.regionfreedvd.net and http://regionhacks.datatestlab.com.

## 3. DESIGN CHALLENGES

Content protection systems must address many technical challenges. Although a complete requirements analysis is beyond the scope of this paper, Figure 4 lists several of the major security and design requirements reflected in our analysis. The feasibility of meeting these requirements will be reviewed in detail at the conclusion of this paper (Section 10).

- Renewability
- Playability
- End-to-End Security
- Cost
- Openness
- Player Diversity
- Migration Path
- Assurance
- Incentives for Security
- Forensic Reporting

**Figure 4: Design challenges for content protection systems.**

## 4. RISK MANAGEMENT FUNDAMENTALS

Although cryptographic algorithms and some other elements used in copy control systems can be extremely secure, other components are much more difficult to protect. For example, determined adversaries will find ways to copy media, modify players, and redistribute data. As a result, we have little optimism that any complete copy protection system will survive unbroken throughout the life of a successful media format. The lack of perfect security does not necessarily support claims that rights holders need to adopt new business models because "copy protection efforts are doomed"[6] and rampant piracy is inevitable.

Risk management approaches have the potential to provide a long-term deterrent without perfect security. Instead of trying to anticipate and prevent every possible attack, risk management systems are designed to respond to dynamic threats and recover from compromises.

Other industries depend on risk management to control security problems that cannot be solved completely. For example, software vendors have largely failed to produce *defect-free programs*, but provide users with patches to address security risks as they are discovered. Similarly, anti-virus programs require frequent updates in order to detect newly-discovered viruses. Although reactive approaches will never eliminate security risks, attacks can be prevented from getting out of control. Without security updates, the Internet as we know it could not exist because each new flaw or virus would be catastrophic.

Financial institutions also rely on risk management techniques. Although credit card networks are based on fundamentally insecure magnetic stripe technology, risk management tools have been able to hold credit card fraud rates below 0.1% of transaction volume.[7] In practice, even lower fraud rates could be achieved by adjusting credit scoring and transaction risk management parameters, but doing so would tend to decrease profits by denying more valid transactions and increasing costs.

Risk management systems are only effective if they provide the ability to detect attacks and to respond. For example, software companies actively seek out information about new viruses and security flaws, then respond by issuing updates. Similarly, credit card companies detect fraud by using neural networks and other risk assessment tools to analyze data collected from point-of-sale terminals. When a high-risk transaction is identified, actions are taken to mitigate the risk, such as declining the transaction, obtaining additional cardholder verification, or suspending the account. Because responses incur costs (such as the loss of customers whose transactions were declined), risk management approaches try to maintain a steady state that balances risks and mitigation costs (see Figure 5).

Content protection systems have several important advantages over credit card security systems. For example, fraud rates considerably higher than 0.1%

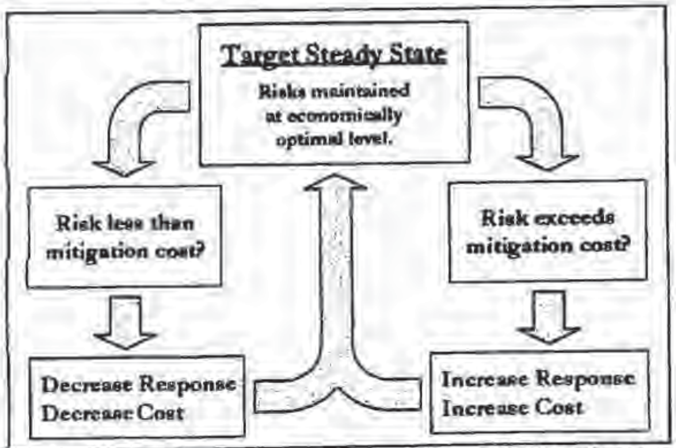


**Figure 5: Using risk management to approach an optimal steady state.**

---

[6] Chmielewski, Dawn, "Andreessen: Copy protection efforts are doomed," The Mercury News, Apr. 9, 2002. (Available on-line from http://www.siliconvalley.com.)

[7] "Fraud Rates Decline with Visa's Innovative, End-to-End Solutions", Visa USA media release, September 2001.

| **Requested Actions** | **Player Information** | **Media Information** | **Output Information** | **User Information** |
|---|---|---|---|---|
| Play | Model/version | Format | Type | Name |
| Copy | Form factor | Recordable | Manufacturer | E-mail address |
| Record | Memory contents | Pre-recorded | Quality/bit rate | Telephone # |
| Export/Convert | Revision status | Capacity | Version | Payment card # |
| Eject | Playback history | Manufacturer | Device keys/certs | Registration # |
| Delete | Serial number | Serial number | Serial number | IP address |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |

**Figure 6: Examples of players information on which risk management decisions can be made.**

are generally tolerable (though undesirable) because piracy represents lost opportunity instead of lost money.[8] Similarly, while stolen credit cards can be used to buy goods that can be fenced, more effort is required to convert stolen intellectual property into cash.

Despite these advantages, content protection technologies must be able to operate without on-line notification and authentication when content is rendered. As a result, risk management systems must be specially designed to enable content owners to detect problems and to respond effectively.

## 5. PROGRAMMABLE SECURITY

Threats against anti-piracy systems are dynamic and unpredictable. Although some existing systems can detect or respond to specific types of attack, approaches that address a limited aspect of the problem (such as decoder compromises) are of little use if attackers can simply target other parts of the system (such as digital outputs). To be effective, content protection systems must have the ability respond effectively to an extremely broad range of threats – including attacks that were not anticipated when the system was originally designed.

Existing anti-piracy systems generally use static decoding processes that are defined as part of the media format and implemented in every player. Of these schemes, some newer ones (such as CPPM[9] used for DVD-Audio) support the revocation of individual players, although it is unclear how compromised devices would be identified. Static systems also generally lack the flexibility required to address security risks beyond the decoder itself, such as compromises of digital output devices or software device drivers. If a static system is widely broken, as occurred with DVD-CSS, the problem cannot be remedied without replacing the installed base of players.

We believe that future formats must be able to mitigate unexpected risks. Instead of implementing the security system solely in the player, much of the content's protection system and decoding software can be ***distributed as part of the content itself.*** Having each title carry its own security logic, policies, and countermeasures makes it no longer necessary to anticipate and prevent all possible attacks when the media format is designed. Deferring security decisions *until* the content is mastered (or, in some cases, decoded) allows security problems to be corrected without changes to the media format or the installed base of players.

The content's protection system and decoding software can be distributed as part of the content itself.

Under this type of security architecture, the player provides an execution environment for the security code that is distributed with the content. The player component would typically be implemented as an interpreter or virtual machine (as used by languages such as Java™ or BASIC). The player would also provide the content's code with access to cryptographic primitives and detailed data about the playback environment, such as the information in Figure 6.

Although the player provides raw information, the content's code controls how this information is used. For example, if a player has marginal security or if the user is making a copy, the content might decide to play at standard quality. High-definition playback could be reserved for players with superior security. If a player is

---

[8] For an interesting economic analysis, see: Liebowitz, Stan, "Policing Pirates in the Networked Age," Policy Analysis No. 438, Cato Institute, May 15, 2002.

[9] "Content Protection for Prerecorded Media Specification", available from the 4C Entity, June 28, 2000.

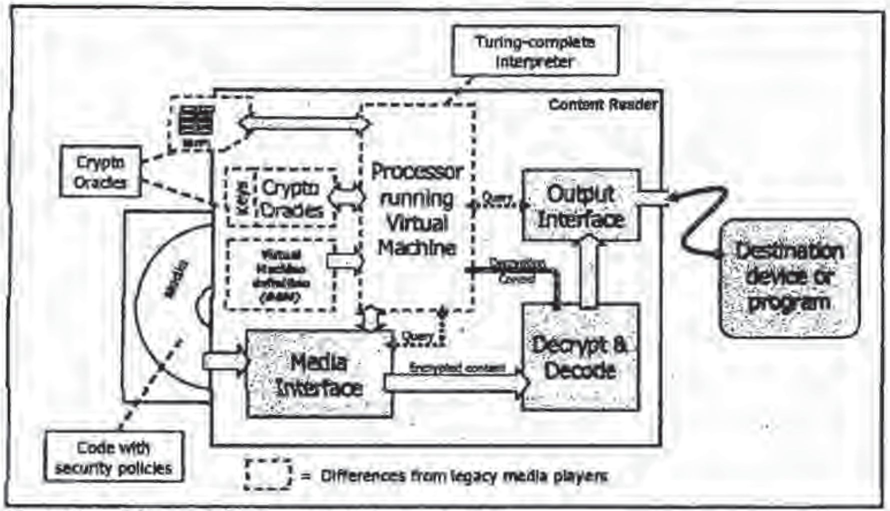


Figure 7: Architecture of a programmable content player.

known to be compromised or cannot be trusted to provide correct information, the content could refuse to play, at least until the player's security is upgraded. Of course, for titles where piracy is not a concern, code could allow unrestricted playback on all players.

The flexibility gained by separating the player design from the security code can improve both security and the user experience. For example, existing systems often allow only system-wide, irreversible, all-or-nothing choices about whether to revoke players with marginal security. In contrast, programmable systems allow flexible responses such as allowing playback at reduced quality, adding user verification steps, or displaying customized warning messages.

Programmable systems can also solve unexpected problems. For example, even though this capability was not planned, a publisher could prevent discs in multi-disc sets from being sold or rented separately by checking for the first disc of the set in the player's history. Greater flexibility can also help with antitrust issues by allowing participants to make their own security decisions. Although this paper focuses on security issues, programmability can also be used for non-security purposes.[10] For example, content-based code can be used to overcome format limitations or provide user interactivity.

## 6. IMPLEMENTATION

Figure 7 outlines the general architecture of a typical programmable content player. The player ROM contains code for an interpreter (virtual machine) instead of the static security policies used by legacy systems. As described previously, the interpreter would also provide the content's code with information about the playback environment as well as cryptographic support. If desired, some keys could be placed on a removable security module, such as a smart card.

The content's code needs to have access to cryptographic functions that use the player's keys, but the code should not have access to the keys themselves. Architectures that do not provide this separation are vulnerable to compromise by poorly or maliciously designed content. Hardware-based players should ideally separate player keys in a separate EEPROM memory that is accessible only by the player's cryptographic module. Software-only decoders would typically store keys in obfuscated form. Drives for use in general-purpose PCs could also include cryptographic keys and support in the drive itself.

Prior to deployment, the playback process needs to be standardized. This effort would include defining the interpreter, the programming interfaces (APIs) that provide the content code with information about the playback environment, and the key management system. Considerable technical expertise is required to produce good specifications, particularly for highly-constrained and complex systems. Although often neglected, careful testing and verification are also necessary to provide high assurance in a design's security.[11]

Compared to legacy designs, hardware-based decoders will tend to use slightly more silicon area. Software-based decoders are likely to incur a modest performance overhead and use slightly more RAM. These differences should be minor, however, when

---

[10] Note that adding simple programmability to a platform is not sufficient for security purposes. For example, existing video game players lack security-related APIs and key management capabilities necessary to enable secure device revocation and forensic marking.

[11] Careful evaluations reduce the chance of unexpected failures and help relying parties understand their risks. Cryptography Research encourages third-party evaluations of all security designs, including our own. For critical systems, testing can exceed the design effort by a factor of 10 or more.

compared to the advances predicted by Moore's Law.[12] The additional storage space required for security code should be negligible given the storage capacities available on modern optical media.

For basic security capabilities, an interpreter capable of 1 MIPS with 128 kilobytes of memory would be minimal but adequate. As with non-programmable systems, a small nonvolatile memory for storing keys and a higher-speed cryptographic module would also be needed. The nonvolatile memory should also include room for carrying software updates, player information, cryptographic certificates, identifiers of revoked devices/media, and historical information about previous media and attached devices. In theory, a basic design should not add more than a few cents to the incremental manufacturing cost of a high-volume hardware-based player,[13] and nothing for a software-only player. Other costs for product vendors include product design and technology licensing, although these are partially offset by transferring responsibility for security policy implementations to rights holders.

More expensive designs could offer better performance, security, and features. For example, players that store and manage their keys and historical data in separate dedicated hardware can offer better tamper resistance. Players with Internet or telephone connectivity could support on-line security verification, downloadable security updates, and alternative business models such as pay-per-view. Secure internal clocks could also enable subscription-based pricing models. Higher-performance systems with video displays could even support general-purpose computing applications such as web browsers, interactive content, or video games.[14]

These features, and virtually all others, could be optional. Manufacturers could add extensions or features to their products and offer them to publishers. The content's code would determine what capabilities are supported and decide whether and how to use them. Even security itself can be optional, since rights holders could control whether products such as unsecured open-source software decoders or disc copiers could decode their content. In practice, coordination between product vendors and rights holders is also important to ensure a consistent and positive customer experience.

While publishers would be responsible for mastering their own content, we expect a market to develop for third-party tools. These tools could range from simple protection systems to full-featured digital rights management systems (DRMs).[15] Vendors would compete to provide publishers with the best features, security, and cost.

Although the content would control its own security, some key management processes should be centralized to help ensure compatibility. This service would provide product manufacturers with certificates describing their products' capabilities, and would provide publishers with information about players. It would also supply keys to enable new products to decode older content (subject to the content's security policies). It would also provide data to publishers so that their content could be decoded by players issued in the future (again, subject to the content's security policies). If desired to stimulate competition, multiple key management services could exist in parallel.

## 7. POINT-TO-POINT VS. END-TO-END SECURITY

The models pursued by the SDMI committee and most other anti-piracy standardization efforts are based on providing point-to-point security. Content is encrypted when it is stored on media or communicated between devices. Each device decrypts the input it receives, decompresses the data, and (for digital outputs) re-encrypts it for the next component. Additional devices decrypt, process, and re-encrypt the content until it is ultimately sent to an analog output. Figure 8 shows an example of a point-to-point system with three devices.

Point-to-point systems are only secure if all supported devices and protocols are secure. For example, if the keys from one device's input are cracked and published on the Internet, other devices will continue to output content encrypted using these keys. Even if content owners are aware of the attack, nothing

---

[12] Every 18 months, the number of transistors per square millimeter is predicted to double, and the cost per transistor will fall by half.

[13] As of July 2002, retail DRAM costs are below 0.03 cents/kilobyte, flash memory prices are below 0.04 cents/kilobyte, and CPU prices are below 10 cents/MHz. Actual costs could be higher if a new chip was required, or lower if the necessary hardware was already available.

[14] It is important to note that programmability is necessary but not sufficient for decoders to support self-protecting content. For example, conventional computers or video game machines would at least require additional software.

[15] The DRM industry is currently struggling due to the difficulty of simultaneously and ubiquitously deploying compatible players and content. Programmable systems can help by eliminating the need for explicit player support for each DRM.

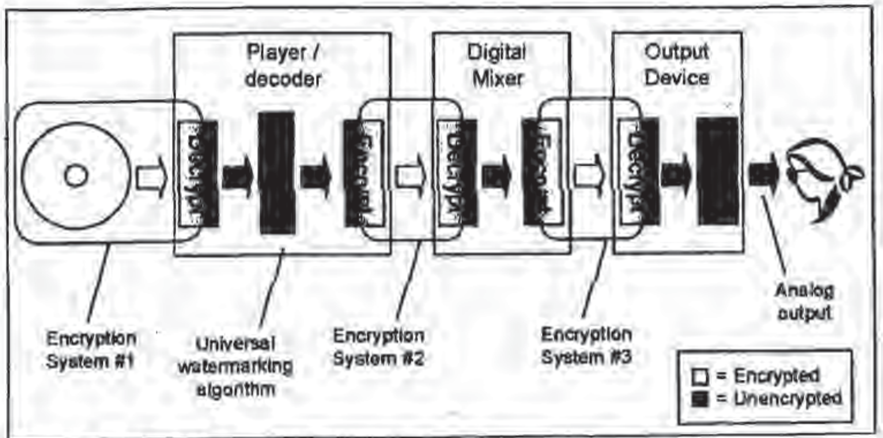


**Figure 8: Point-to-point content protection system.**

can be done to address the problem without losing compatibility with all fielded devices.

Although some existing schemes allow the revocation of individual player devices, player revocation is generally not effective against downstream attacks. For example, if the output device in Figure 8 is compromised, the content cannot prevent intermediate devices from using the compromised keys. In fact, the player device is unaware of how the content will ultimately be used. Player revocation features are also of limited use unless there is a practical way to detect compromises and respond to situations where a large number of devices share a security flaw.

Systems providing end-to-end validation can provide much better risk management capabilities than those with only point-to-point security. Figure 9 diagrams the operation of a sample system with end-to-end security using the program-based approaches described previously. Although links between devices are still encrypted individually, the initial decoder device validates how the content will be used downstream.

End-to-end validation can be implemented by having the player/decoder provide an interface through which the content's security code can identify and query downstream objects. The code can use this information to control whether and how playback would proceed and to deliver security parameters or even security code to downstream devices.

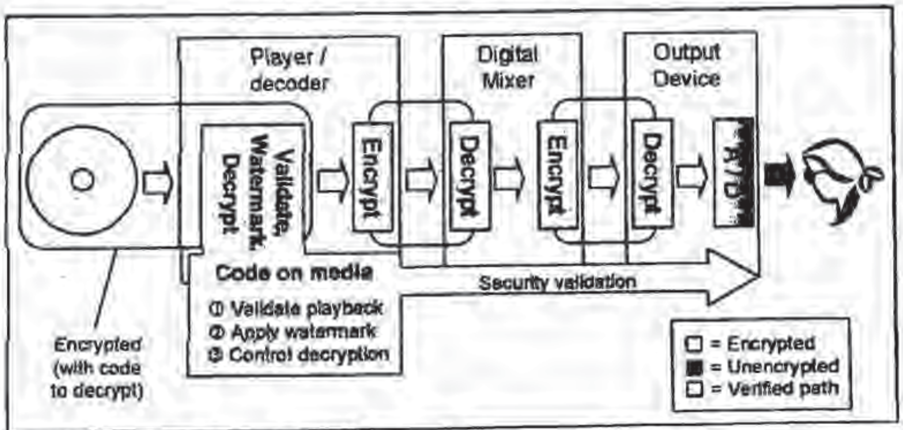


**Figure 9: End-to-end content protection system.**

In Figure 9, the plaintext (decrypted) content does not leave the validated environment until the final analog-to-digital conversion. Compromises prior to the analog conversion can be handled using the content-controlled programmable risk management approaches described in Section 5, while forensic marking techniques (see Sections 8 and 9) can help prevent piracy from analog outputs.

In general, we believe that point-to-point designs are unlikely to provide a long-term deterrent in major deployments due to their lack of risk management capabilities. End-to-end systems are not necessarily any less likely to be broken, but are likely to prove much more effective over the long-term because recovery is possible from a much broader array of compromises.

## 8. PUBLIC (CONVENTIONAL) WATERMARKING

Watermarks have been proposed as a way to detect and control copying. For example, the SDMI committee planned to use an audio watermark to convey a "do not copy" signal to recording devices. This design implies a "public" watermarking system, consisting of a mark embedding algorithm (which can be public or private) and a public detection algorithm. The detection algorithm is assumed to be public (known to attackers) because it must be standardized and deployed in large numbers of recording devices, some of which will eventually be reverse engineered.[16]

Although secure public watermarking systems would be enormously useful in combating piracy, there are convincing arguments that they are impossible to construct for audio, video, images, and other normal content. The basic challenge is that knowledge of the detector allows attackers to determine when the mark has been removed. For example, a simple automated attack that will break all schemes we know about is to use successive approximation (also called sensitivity analysis) to construct unwatermarked versions of marked content by repeatedly making tiny changes until the mark is no longer detected (see Figure 10).[17]

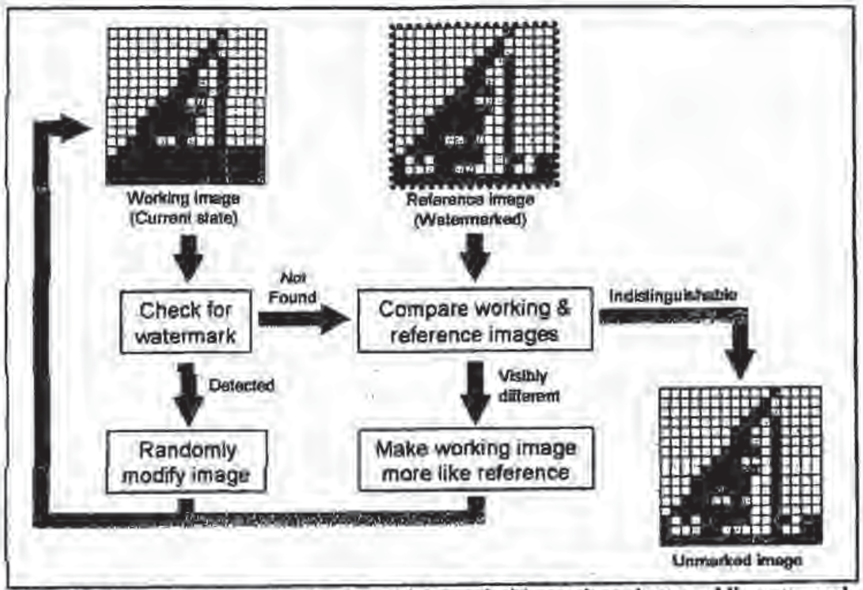


**Figure 10: Successive approximation (sensitivity analysis) attack against a public watermark.**

In addition to security concerns, current watermarking proposals are computationally complex, making them expensive to embed and to detect. Other common problems include distracting artifacts and the inability to survive common transformations such as cropping and compression. Although some progress is being made at improving robustness and efficiency, we are not optimistic that a practical and secure public watermarking scheme is possible.

## 9. FORENSIC MARKING

For effective risk management, publishers must be able to respond to attacks. Although programmable security capabilities can provide a flexible response mechanism, appropriate responses require knowledge about the specific equipment and processes used to make pirated copies. Methods used to convey this information need to be secure, efficient, and respect users' privacy.

Because players must be able to operate off-line, the only practical and effective channel for reporting information is the output content itself. Although conventional watermarks could theoretically be adapted for this purpose, forensic marks provide a practical and provably secure alternative. Forensic marks embed identifying and diagnostic information in outputs, but do not use a fixed detector. As a result, they are able to avoid the security problems with conventional watermarks, but cannot be used in systems such as SDMI where the detection algorithm must be standardized and deployed widely.

To embed each bit of a typical forensic mark, the player device decrypts and outputs one of two (or more) versions for a portion of the content (see Figure 11). From even a heavily-degraded analog recording, the embedded data can be recovered by determining which of the versions is present. Because the detection process is not fixed, each mark bit can be represented by virtually any difference in the output. If the decoding process is controlled by content-specific security code, this code can choose what to output and can also generate decryption keys to secure the selection. The actual information that is encoded in the forensic marks could include any data available during playback, such as the parameters listed in Figure 6 (page 7).

---

[16] In practice, many systems can often be broken without even reverse engineering the detector. For example, see: Craver, Scott et al., "Reading Between the Lines: Lessons from the SDMI Challenge", Proceedings of 10th USENIX Security Symposium, August 2001.

[17] For more information about this attack and several others, see Cox, I., Miller, M., and Bloom, J., Digital Watermarking, Morgan Kaufmann Publishers, 2002, pages 307-317.

In a simple example using video, the media might carry two versions (polymorphs) for a small portion of each of 500 video frames. During playback, the content's security code first obtains data identifying the player device and any output devices. The code uses this data to select which *version of each* polymorphic frame to decrypt. Given a recording of the decoded content, the publisher can determine which version of each marked frame is present, and use the recovered data to identify the devices used to make the copy.

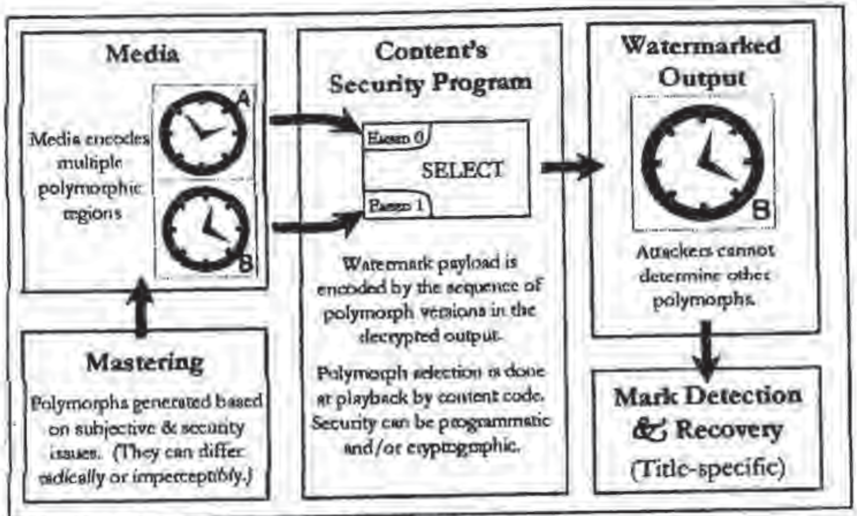


**Figure 11: Content-controlled embedding of a forensic mark.**

Forensic marks can be both provably secure and provably robust. Because no constraints are placed on the variations (polymorphisms) in the content, knowledge of one does not enable attackers to determine others. The polymorphs are stored on the media or generated on-the-fly by the content's code, and can be protected using conventional cryptographic or programmatic security measures. The locations of variations can also be concealed securely by encrypting portions of the decoding software. Without knowledge of what variations are present or where they are located, attackers cannot reliably remove forensic marks without destroying the content. (See Figure 12 for an example.)

The plaintext content is divided into portions $P_1..P_n$. A randomly-selected portion $P_i$ ($1 \leq i \leq n$) is modified to create an alternate version $P_i'$ such that the change cannot be identified from the context ($P_1..P_{i-1}$ and $P_{i+1}..P_n$). Portions $P_1..P_n$ and $P_i'$ are encrypted with random keys $K_1..K_n$ and $K_i'$ then stored on the media in random order. *A first decoding program* $D_1$ is constructed that includes keys $K_1..K_n$ and indexes for locating the encrypted $P_1..P_n$ on the media. A second decoding program $D_2$ is constructed with $K_1..K_{i-1}$, $K_i'$, $K_{i+1}..K_n$ and indexes to $P_1..P_{i-1}$, $P_i'$, $P_{i+1}..P_n$.

Programs $D_1$ and $D_2$ are encrypted with program keys $K_{P1}$ and $K_{P2}$, respectively, and stored on the media. Finally, the values of $K_{P1}$ and $K_{P2}$ are placed on the media encrypted so that the set of players that should embed the bit '0' in the mark can determine $K_{P1}$ (and only $K_{P1}$), while all other valid players (which embed '1') can only recover $K_{P2}$.

A player decrypts either $D_1$ or $D_2$ using $K_{P1}$ or $K_{P2}$. Because $D_1$ decrypts the content with $P_i$ while $D_2$ decrypts with $P_i'$, the value of the marked bit can be recovered by analyzing the output.

An attacker with either $D_1$ or $D_2$ (or their outputs) cannot determine which portion has multiple versions or what the differences are. As a result, the adversary cannot reliably destroy the mark without also destroying the content so extensively that all possible changes become undetectable (i.e., completely obliterating the work). An adversary with both $D_1$ and $D_2$ can produce an output containing both $P_i$ and $P_i'$ or that omits $P_i$ and $P_i'$, but this reveals even more information to the publisher, notably that the copy was made by combining outputs from at least one device in each group.

**Figure 12: Example of a provably-secure, provably-robust forensic mark.**

Because content-specific code can control the decryption process, publishers can choose during the mastering process what data will be encoded in each mark, where marks will be placed, and how marks are encoded. For example, variations can be chosen to accommodate artistic or subjective requirements. Marking can also be disabled if piracy is not a concern.

When a pirated copy is recovered, mark data can be extracted and used to master future content so that it cannot be played or decrypted using the same compromised or misused devices. Copies produced by combining multiple outputs can even be traced (see Figure 13).[18] This detection and revocation capability forces pirates to put their equipment at risk and can provide evidence for prosecution. Finally, because people are more likely to misbehave in situations where they feel anonymous, simply making users aware that

---

[18] For a detailed analysis, see: Boneh, D., and Shaw, J., "Collusion-Secure Fingerprinting for Digital Data", IEEE Transactions on Information Theory, Vol 44, No. 5, 1998, pp. 1897-1905.

> "Absolute anonymity breeds absolute irresponsibility."
> —Scott McNealy, Chairman & CEO, Sun Microsystems

copies are traceable is expected to reduce piracy. At the same time, forensic marks avoid the privacy concerns associated with other data collection approaches because no information is revealed about users who do not redistribute copies.

General information gathered from forensic marks can also help publishers make appropriate risk management decisions. For example, if piracy using a particular software decoder becomes widespread, a content owner might prevent it from decoding future content at high resolution until users install a security upgrade.

## 10. REVIEW OF DESIGN OBJECTIVES AND REQUIREMENTS

Figure 4 in Section 3 lists major requirements and objectives for content protection systems. This section reviews these issues and the feasibility of addressing them using self-protecting content with forensic marks.

- **Renewability** – Security must be reestablished after individual devices are compromised or flaws are found in product designs.

  No limitations are imposed on number of compromises or attacks that can be survived. Many compromises can be repaired using code updates. Unaffected products are not impacted.

If $k$ out of $N$ decoders collude to try to remove a forensic mark, there are $\binom{N}{k}$ possible sets of colluders. A set of colluders can be excluded if no set members could decode the observed version of a polymorph. If each version of each polymorph can be decrypted by an independent random 50% of decoders, each polymorph in the output excludes $(\frac{1}{2})^k = 2^{-k}$ of the collusion sets. If a total of $p$ polymorphs are present, the expected number of non-excluded *collusion sets is* $(1-2^{-k})^p\binom{N}{k}$.

For example, a 90-minute movie at 30 frames/second has 162,000 frames. For 1% of the frames ($p$=1620), two polymorphs are included. Even if an adversary produces a pirate copy by combining outputs from 4 decoders ($k$=4) chosen from a population of 1 billion decoders ($N$=$10^9$), the content owner can identify all of the compromised devices with probability >99.9999999%, since the expected number of ambiguous collusion sets is:

$$(1-2^{-4})^{1620}\binom{10^9}{4} < \left(\tfrac{15}{16}\right)^{1620}(10^9)^4 = e^{1620(\ln 15-\ln 16)+36\ln(10)} < e^{-21.6} < 4\times10^{-10}.$$

**Figure 13: Simple traitor tracing (collusion detection) example.**

- **Playability** – All valid players must be able to play all valid content, subject to security policies.

  Operation is fully configurable by publisher, but security would normally be hidden and automatic. Flexibility allows publishers to block unauthorized actions while minimizing any impact on legitimate users.

- **End-to-End Security** – Content should be protected through the entire distribution and playback process.

  Security code can validate all information available during the playback sequence, including decoder types, media types, software device drivers, devices connected to digital outputs, etc. Forensic marks deter copying from analog and other outputs.

- **Cost** – Cost should be minimized.

  Modest impact on player complexity; manufacturing cost today should be less than costs for CSS when DVD was introduced. Effort to develop/procure security code would increase content mastering costs. Fixed costs include administration, technology licensing, player engineering, and standards development.

- **Openness** – Because implementations will eventually be reverse engineered, security must not rely on the secrecy of the system's design.

  All system design documents could be made public; only players' production keys need to be secret.

- **Player Diversity** – Security must be provided across a broad range of decoding devices.

  Support for all player types is practical, including those that are software-based, portable, and off-line. Future player types and security features can be supported in future content. Because publishers/artists can decide where their content will be played, content code can range widely in features and security policies.

- **Migration Path** – Transitions from one format to another should be as smooth as possible.

  To support migration from insecure designs, players can support both legacy and self-protecting content formats. Legacy standards can be implemented in updateable code running on the player's interpreter. Upgrades and transitions from programmable formats can be done by adding appropriate code to content.

- **Assurance** – System-level designs must provide high assurance of security, while assuming that individual implementations may be insecure.[19]

  System design assurance is only limited by the standards process, quality of documentation, and third party

---

[19] Security products are uniquely difficult to evaluate because security flaws are invisible during normal operation and vendor claims are notoriously unreliable. Careful due diligence of all security claims (including our own) is strongly encouraged.

evaluations. Cryptographic components and forensic marking can be provably secure. Security flaws in content code do not affect other titles. Player flaws can affect older content, but can be avoided or repaired in new content.

- **Incentives for Security** – *Vendors must have tangible market-based incentives to ensure security, even after a format has been adopted.*

  Programmable designs give manufacturers an ongoing incentive to invest in security, since publishers will trust products with better security with their most compelling, highest-quality, and newest content.

- **Forensic Reporting** – *It should be possible to identify the specific devices and methods used by pirates.*

  Forensic marks allow content to embed arbitrary information about the decoding process in the output. Publishers can recover this data from even a degraded analog copy and use it to revoke pirates' equipment, improve the security of new content, and prosecute pirates.

In addition to these design issues above, some attacks cannot be prevented completely by any player or media technology. Although these will always remain sources of piracy, risk management approaches can provide useful responses:

- **Media cloning** – No technology can distinguish between original media and a perfect copy.

  Although players can detect user-recordable media and reject media with revoked IDs, law enforcement efforts will be required to stop professional pirates who obtain access to equipment for making exact copies to non-consumer-recordable media. (Proprietary media features may help in the short term, but will eventually be reverse engineered or circumvented by professional pirates.)

- **Analog Recording** – No technology can eliminate recording from analog or unprotected outputs.

  Although general-purpose recording devices will always be able to record from analog outputs, forensic marks can trace copies back to specific devices, which can then be revoked.

- **File Sharing** – No technology can eliminate copying of content that has had its protection removed.

  Once content has been converted to a format that lacks security features, it can be redistributed, e.g. via computer networks. Although player security features and forensic marking may help deter this piracy or trace its source, we do not suggest that improvements in player security alone will solve the problem of piracy over Internet file sharing networks.

## 11. CONCLUSIONS

It is impossible to predict the specific attacks and threats that anti-piracy systems will face. Conventional static security approaches are ineffective because they lack the flexibility required to respond to unexpected problems. In contrast, programmable systems eliminate the need to anticipate all future threats by separating critical security design choices from the media format and player design. When failures occur, as we expect they inevitably will, publishers can mitigate their risk by revising security systems and policies without losing compatibility with the installed base of players.

> "Failure is only the opportunity to begin again more intelligently."
> — Henry Ford

Programmable systems can adapt and evolve as technical advances yield new threats and opportunities. This provides content owners with the ability to respond and to recover from attacks that would have otherwise been catastrophic. The intended result is a chess game of pirate attacks and publisher countermeasures. Newer content will benefit from newer security measures, while older content is more likely to be pirated. Piracy will not be eliminated, but programmatic responses such as forensic marking, equipment revocation, and code upgrades can provide an ongoing deterrent by increasing the risk, cost, and effort of piracy.

Publishers have been effective at lobbying, but have not presented a long-term technical strategy. While new anti-piracy systems could be far more effective than any in use today, investments in security have been inadequate relative to the major economic threat posed by piracy.

Efforts to improve security will require strong technical leadership. Otherwise, standards efforts will tend to degenerate into unwieldy and ineffective committees with short-term focus. Leadership is also needed to prevent ineffective proposals from wasting time and momentum, to verify that security needs are met before products ship, and to help secure designs succeed in the marketplace. We conclude that only rights holders can provide this leadership; no other participants have the motivation, expertise, or resources to ensure the deployment of effective anti-piracy technologies.

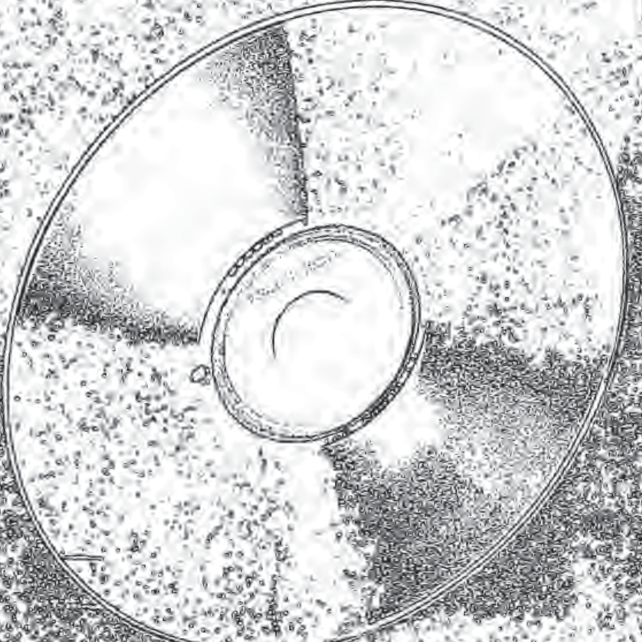
* * *

PROCEEDINGS OF THE IEEE

THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC. JULY 1999

Special Issue

# IDENTIFICATION & PROTECTION OF MULTIMEDIA INFORMATION



Papers on: Information Hiding
* Watermarking Techniques
* Digital & Video Perceptual Watermarks
* Side-Information Communication
* Copyright Protection
* Tamper-Proofing & Authentication
* Electronic Distribution Protection
* Digital Multimedia Watermarks
* Unique Identifiers
* Intellectual Property Protection
* Access Control * Open Net Delivery
and DVD Video

*plus*

Classic 1898 Paper: Nikola Tesla:
"High Frequency Oscillators for
Electro-Therapeutic & Other Purposes."

*Predictive Paper: Communication
Spectra by the Wholesale-2012 A.D.
by Estill Green*

The Electrical Century: Viewing
Television's History

# Multimedia Watermarking Techniques

FRANK HARTUNG, STUDENT MEMBER, IEEE, AND MARTIN KUTTER

*Invited Paper*

*Multimedia watermarking technology has evolved very quickly during the last few years. A digital watermark is information that is imperceptibly and robustly embedded in the host data such that it cannot be removed. A watermark typically contains information about the origin, status, or recipient of the host data. In this tutorial paper, the requirements and applications for watermarking are reviewed. Applications include copyright protection, data monitoring, and data tracking. The basic concepts of watermarking systems are outlined and illustrated with proposed watermarking methods for images, video, audio, text documents, and other media. Robustness and security aspects are discussed in detail. Finally, a few remarks are made about the state of the art and possible future developments in watermarking technology.*

***Keywords*— Audio, image, multimedia, review, video, watermarking.**

## I. INTRODUCTION

Multimedia production and distribution, as we see it today, is all digital, from the authoring tools of content providers to the receivers. The advantages of digital processing and distribution, like noise-free transmission, software instead of hardware processing, and improved reconfigurability of systems, are all well known and obvious. Not so obvious are the disadvantages of digital media distribution. For example, from the viewpoint of media producers and content providers, the possibility for unlimited copying of digital data without loss of fidelity is undesirable because it may cause considerable financial loss. Digital copy protection or copy prevention mechanisms are only of limited value because access to cleartext versions of protected data must at least be granted to paying recipients which can then produce and distribute illegal copies. Technical attempts to prevent copying have in reality always been circumvented.

One remaining method for the protection of intellectual property rights (IPR) is the embedding of digital watermarks into multimedia data. The watermark is a digital code unremovably, robustly, and imperceptibly embedded in the host data and typically contains information about origin, status, and/or destination of the data. Although not directly used for copy protection, it can at least help identifying source and destination of multimedia data and, as a "last line of defense," enable appropriate follow-up actions in case of suspected copyright violations.

While copyright protection is the most prominent application of watermarking techniques, others exist, including data authentication by means of fragile watermarks which are impaired or destroyed by manipulations, embedded transmission of value added services within multimedia data, and embedded data labeling for other purposes than copyright protection, such as data monitoring and tracking. An example for a data-monitoring system is the automatic registration and monitoring of broadcasted radio programs such that royalties are automatically paid to the IPR owners of the broadcast data.

The development of watermarking methods involves several design tradeoffs. Watermarks should be robust against standard data manipulations, including digital-to-analog conversion and digital format conversion. Security is a special concern, and watermarks should resist even attempted attacks by knowledgeable individuals. On the other hand, watermarks should be imperceptible and convey as much information as possible. In general, watermark embedding and retrieval should have low complexity because for various applications, real-time watermarking is desirable. All of these (partly contradicting) requirements and the resulting design constraints will be discussed in more detail throughout the paper.

The paper is organized as follows. Section II gives an introductory explanation of the terms used, as well as a few remarks about the historical aspects of watermarking. In Section III, common design requirements and principles are explained that apply to all watermarking techniques, independent of the actual application. Sections IV–VII review various watermarking techniques that have been proposed for formatted text data, images, video, and audio, respectively. Watermarking of other media, including three dimensional (3-D) data and 3-D animation parameters, is discussed in Section VIII. Section IX gives detailed insight

---

Manuscript received October 20, 1997; revised March 26, 1998.

F. Hartung was with the Telecommunications Laboratory, University of Erlangen–Nuremberg, 91058 Erlangen, Germany. He is now with Ericsson Eurolab, Research Department, 52134 Herzogenrath, Germany.

M. Kutter is with Signal Processing Laboratory, Swiss Federal Institute of Technology, 1015 Lausanne, Switzerland.

Publisher Item Identifier S 0018-9219(99)05174-9.

into security issues, namely attacks against watermarks, and shows the relations between watermarking and cryptology. In Section X, we extrapolate the recent development of watermarking technology and watermarking applications and try to forecast future trends. Section XI summarizes and concludes this paper on multimedia watermarking techniques.

## II. STEGANOGRAPHY AND WATERMARKING—HISTORY AND TERMINOLOGY

### *A. History*

The idea to communicate secretly is as old as communication itself. First stories, which can be interpreted as early records of covert communication, appear in the old Greek literature, for example, in Homer's *Iliad*, or in tales by Herodotus. The word "steganography," which is still in use today, derives from the Greek language and means covert communication. Kobayashi [67] and Petitcolas *et al.* [99] have investigated the history of covert communication in great detail, including the broad use of techniques for secret and covert communication before and during the two World Wars, and steganographic methods for analog signals. Although the historical background is very interesting, we do not cover it here in detail. Please refer to [67] and [99] for an in-depth investigation of historic aspects.

Paper watermarks appeared in the art of handmade papermaking nearly 700 years ago. The oldest watermarked paper found in archives dates back to 1292 and has its origin in Fabriano, Italy, which is considered the birthplace of watermarks. At the end of the thirteenth century, about 40 paper mills were sharing the paper market in Fabriano and producing paper with different format, quality, and price. They produced raw, coarse paper which was smoothed and postprocessed by artisans and sold by merchants. Competition not only among the paper mills but also among the artisans and merchants was very high, and it was difficult to keep track of paper provenance and thus format and quality identification. The introduction of watermarks helped avoiding any possibility of confusion. After their invention, watermarks quickly spread over Italy and then over Europe, and although originally used to indicate the paper brand or paper mill, they later served as indication for paper format, quality, and strength and were also used to date and authenticate paper. A nice example illustrating the legal power of watermarks is a case in 1887 in France called "Des Decorations" [41]. The watermarks of two letters, presented as pieces of evidence, proved that the letters had been predated and resulted in considerable sensation and, in the end, in the resignation of President Grévy. For more information on paper watermarks, watermark history, and related legal issues, please refer to [144], an extensive listing of over 500 references.

The analogy between paper watermarks, steganography, and digital watermarking is obvious, and in fact, paper watermarks in money bills or stamps [135] actually inspired the first use of the term watermarking in the context of digital data.

The idea of digital image watermarking arose independently in 1990 [131], [132] and around 1993 [20], [136]. Tirkel *et al.* [136] coined the word "water mark" which became "watermark" later on. It took a few more years until 1995/1996 before watermarking received remarkable attention. Since then, digital watermarking has gained a lot of attention and has evolved very quickly, and while there are a lot of topics open for further research, practical working methods and systems have been developed. In this paper, we introduce the concepts and illustrate them with some of the work that has been published. While attempting to be as complete as possible, we can still only give a rough overview.

### *B. Terminology*

Today, we are of course concerned with digital communication. As in classical analog communication, also in digital communication there is interest for methods that allow the transmission of information hidden or embedded in other data. While such techniques often share similar principles and basic ideas, there are also important distinguishing features, mainly in terms of robustness against attacks. Several names have been coined for such techniques. However, the terms are often confused, and therefore it is necessary to clarify the differences.

*Steganography* stands for techniques in general that allow secret communication, usually by embedding or hiding the secret information in other, unsuspected data. Steganographic methods generally do rely on the assumption that the existence of the covert communication is unknown to third parties and are mainly used in secret point-to-point communication between trusting parties. As a result, steganographic methods are in general not robust, i.e., the hidden information cannot be recovered after data manipulation.

*Watermarking*, as opposed to steganography, has the additional notion of robustness against attacks. Even if the existence of the hidden information is known it is difficult—ideally impossible—for an attacker to destroy the embedded watermark, even if the algorithmic principle of the watermarking method is public. In cryptography, this is known as *Kerkhoffs law*: a cryptosystem should be secure, even if an attacker knows the cryptographic principles and methods used but does not have the appropriate key [117]. A practical implication of the robustness requirement is that watermarking methods can typically embed much less information into host data than steganographic methods. Steganography and watermarking are thus more complementary than competitive approaches. In the remainder of this paper, we focus on watermarking methods and not on steganographic methods in general. For an overview of steganographic methods the reader is referred to [67], [99], and [124].

*Data hiding* and *data embedding* are used in varying contexts, but they do typically denote either steganography or applications "between" steganography and watermarking, which means applications where the existence of the embedded data are publicly known, but there is no need to protect it. This is typically the case for the embedded transmission of auxiliary information or services [125] that are publicly available and do not relate to copyright protection or conditional access functionalities.

*Fingerprinting* and *labeling* are terms that denote special applications of watermarking. They relate to copyright protection applications where information about originator and recipient of digital data is embedded as watermarks. The individual watermarks, which are unique codes out of a series of codes, are called "fingerprints" or "labels."

*Bit-stream watermarking* is sometimes used for data hiding or watermarking of compressed data, for example, compressed video.

The term *embedded signatures* has been used instead of "watermarking" in early publications. Because it potentially leads to confusion with cryptographic digital signatures [117], it is usually not used anymore. Cryptographic signatures serve for authentication purposes. They are used to detect alterations of the signed data and to authenticate the sender. Watermarks, however, are only in special applications used for authentication and are usually designed to *resist* alterations and modifications.

*Visible watermarks*, as the name says, are visual patterns, like logos, which are inserted into or overlaid on images (or video), very similar to visible paper watermarks. However, the name is confusing since visible watermarks are not watermarks in the sense of this paper. Visible watermarks are mainly applied to images, for example, to visibly mark preview images available in image databases or on the World Wide Web in order to prevent people from commercial use of such images. A visible watermarking method devised by Braudaway *et al.* [16] combines the watermark image with the original image by modifying the brightness of the original image as a function of the watermark and a secret key. The secret key determines pseudorandom scaling values used for the brightness modification in order to make it difficult for attackers to remove the visible mark.

## III. Digital Watermarking

### *A. Requirements*

The basic requirements in watermarking apply to all media and are very intuitive.

1) A watermark shall convey as much information as possible, which means the watermark data rate should be high.
2) A watermark should in general be secret and should only be accessible by authorized parties. This requirement is referred to as security of the watermark and is usually achieved by the use of cryptographic keys.
3) A watermark should stay in the host data regardless of whatever happens to the host data, including all possible signal processing that may occur, and including all hostile attacks that unauthorized parties may attempt. This requirement is referred to as robustness of the watermark. It is a key requirement for copyright protection or conditional access applications, but less important for applications where the watermarks are not required to be cryptographically secure, for example, for applications where watermarks convey public information.
4) A watermark should, though being unremovable, be imperceptible.

Depending on the media to be watermarked and the application, this basic set of requirements may be supplemented by additional requirements.

1) Watermark recovery may or may not be allowed to use the original, unwatermarked host data.
2) Depending on the application, watermark embedding may be required in real time, e.g., for video fingerprinting. Real-time embedding again may, for complexity reasons, require compressed-domain embedding methods.
3) Depending on the application, the watermark may be required to be able to convey arbitrary information. For other applications, only a few predefined watermarks may have to be embedded, and for the decoder it may be sufficient to check for the presence of one of the predefined watermarks (hypothesis testing).

In the following, a few of the mentioned requirements and the resulting design issues are highlighted in more detail.

*1) Watermark Security and Keys:* If security, i.e., secrecy of the embedded information, is required, one or several secret and cryptographically secure keys have to be used for the embedding and extraction process. For example, in many schemes, pseudorandom signals are embedded as watermarks. In this case, the description and the seed of the pseudorandom number generator may be used as key. There are two levels of secrecy. In the first level, an unauthorized user can neither read or decode an embedded watermark nor can he detect if a given set of data contains a watermark. The second level permits unauthorized users to detect if data are watermarked, however, the embedded information cannot be read without having the secret key. Such schemes can, for example, embed two watermarks, one with a public key and the other with a secret key. Alternatively, a scheme has been proposed which combines one or several public keys with a private key and embeds one combined public/private watermark, rather than several watermarks [48]. When designing an overall copyright protection system, issues like secret key generation, distribution, and management (possibly by trusted third parties), as well as other system integration aspects have to be considered.

*2) Robustness:* In the design of any watermarking scheme, watermark robustness is typically one of the main issues, since robustness against data distortions introduced through standard data processing and attacks is a major requirement. Standard data processing includes all data manipulation and modification that the data might undergo in the usual distribution chain, such as data editing, printing, enhancement, and format conversion. "Attack" denotes data manipulation with the purpose of impairing, destroying, or removing the embedded watermarks. Section IX-B below revisits attacks and gives remedies that help to make watermarks attack resistant.

Although it is possible to design robust watermarking techniques, it should be noted that a watermark is only robust as long as it is not public, which means as long as it cannot be read by everyone. If watermark detector principle and key are public, and even if only a "black-box" watermark detector is public, the watermark is vulnerable to attacks [28], [64]. Hence, public watermarks, as sometimes proposed in the literature, are not robust unless every receiver uses a different key. This however is difficult in practice and gives rise to collusion attacks.

*3) Imperceptibility:* One of the main requirements for watermarking is the perceptual transparency. The data embedding process should not introduce any perceptible artifacts into the host data. On the other hand, for high robustness, it is desirable that the watermark amplitude is as high as possible. Thus, the design of a watermarking method always involves a tradeoff between imperceptibility and robustness. It would be optimal to embed a watermark just below the threshold of perception. However, this threshold is difficult to determine for real-world image, video and audio signals. Several measures to determine objectively perceived distortion and the threshold of perception have been proposed for the mentioned media [75]. However, most of them are still not perfect enough to replace human viewers or listeners who judge the visual or audio fidelity through blind tests. Thus, in the design of watermarking systems, it is usually necessary to do some testing with volunteers. The second problem occurs in combination with post watermarking processing, which might result in an amplification of the embedded watermark and make it perceptible. An example is zooming of watermarked images, which often makes the embedded watermarks visible, or contrast enhancement, which may amplify highly frequent watermark patterns that are otherwise invisible.

*4) Watermark Recovery With or Without the Original Data:* Watermark recovery is usually more robust if the original, unwatermarked data are available. Further, availability of the original data set in the recovery process allows the detection and inversion of distortions which change the data geometry. This helps, for example, if a watermarked image has been rotated by an attacker. However, access to the original data is not possible in all cases, for example, in applications such as data monitoring or tracking. For other applications, like video watermarking, it may be impractical to use the original data because of the large data volume, even if it is available. It is, however, possible to design watermarking techniques that do not need the original for watermark extraction. Most watermarking techniques perform some kind of modulation in which the original data set is considered a distortion. If this distortion is known or can be modeled in the recovery process, explicitly designed techniques allow its suppression without knowledge of the original. In fact, most recent methods do not require the original for watermark recovery. In some publications, such techniques are called "blind" watermarking techniques [2], [1].

*5) Watermark Extraction or Verification of Presence for a Given Watermark:* In the literature, two different types of watermarking systems can be found: systems that embed a specific information or pattern and check the existence of the (known) information later on in the watermark recovery—usually using some sort of hypothesis testing—and systems that embed arbitrary information into the host data.

The first type, verification of the presence of a known watermark, is sufficient for most copyright-protection applications.

The second type, embedding of arbitrary information, is, for example, useful for image tracking on the Internet with intelligent agents where it might not only be of interest to discover images, but also to classify them. In such cases, the embedded watermark can serve as an image identification number. Another example where arbitrary information has to be embedded are applications for video distribution where, e.g., the serial number of the receiver has to be embedded.

Although most presented methods or systems are designed for either watermark extraction or verification of presence for a given watermark, it should be noted that in fact both approaches are inherently equivalent. A scheme that allows watermark verification can be considered as a 1-bit watermark recovery scheme, which can easily be extended to any number of bits by embedding several consecutive "1-bit watermarks." The inverse is also true: a watermark recovery scheme can be considered as a watermark verification scheme assuming the embedded information is known.

### *B. Basic Watermarking Principles*

The basic idea in watermarking is to add a watermark signal to the host data to be watermarked such that the watermark signal is unobtrusive and secure in the signal mixture but can partly or fully be recovered from the signal mixture later on if the correct cryptographically secure key needed for recovery is used.

To ensure imperceptibility of the modification caused by watermark embedding, a perceptibility criterion of some sort is used. This can be implicit or explicit, host data adaptive or fixed, but it is necessary. As a consequence of the required imperceptibility, the individual samples (e.g., pixels or transform coefficients) that are used for watermark embedding can only be modified by an amount relatively small to their average amplitude.

To ensure robustness despite the small allowed changes, the watermark information is usually redundantly distributed over many samples (e.g., pixels) of the host data, thus providing a "holographic" robustness, which means that the watermark can usually be recovered from a small fraction of the watermarked data, but the recovery is more robust if more of the watermarked data are available for recovery.

As said before, watermark systems do in general use one or more cryptographically secure keys to ensure security against manipulation and erasure of the watermark.

There are three main issues in the design of a watermarking system.

Fig. 1. Generic digital watermarking scheme.

Fig. 2. Generic watermark recovery scheme.

1) Design of the watermark signal $W$ to be added to the host signal. Typically, the watermark signal depends on a key $\mathcal{K}$ and watermark information $I$

$$W = f_0(I, \mathcal{K}). \tag{1}$$

Possibly, it may also depend on the host data $X$ into which it is embedded

$$W = f_0(I, \mathcal{K}, X). \tag{2}$$

2) Design of the embedding method itself that incorporates the watermark signal $W$ into the host data $X$ yielding watermarked data $Y$

$$Y = f_1(X, W). \tag{3}$$

3) Design of the corresponding extraction method that recovers the watermark information from the signal mixture using the key and with help of the original

$$\hat{I} = g(X, Y, \mathcal{K}) \tag{4}$$

or without the original

$$\hat{I} = g(Y, \mathcal{K}). \tag{5}$$

The first two issues, watermark signal design and watermark signal embedding, are often regarded as one, specifically for methods were the embedded watermark is host signal adaptive.

Figs. 1 and 2 illustrate the concept. Fig. 1 shows the generic watermarking scheme for the embedding process. The input to the scheme is the watermark, the host data, and an optional public or secret key. The host data may, depending on the application, be uncompressed or compressed, however, most proposed methods work on uncompressed data. The watermark can be of any nature, such as a number, text, or an image. The secret or public key is used to enforce security. If the watermark is not to be read by unauthorized parties, a key can be used to protect the watermark. In combination with a secret or a public key, the watermarking techniques are usually referred to as secret and public watermarking techniques, respectively. The output of the watermarking scheme are the modified, i.e., watermarked, data. The generic watermark recovery process is depicted in Fig. 2. Inputs to the scheme are the watermarked data, the secret or public key, and, depending on the method, the original data and the original watermark. The output of the watermark recovery process is either the recovered watermark or some kind of confidence measure indicating how likely it is for the given watermark at the input to be present in the data under inspection.

Many proposed watermarking schemes use ideas borrowed from spread-spectrum radio communications [25], [43], [101]. They embed a watermark by adding a pseudonoise (PN) signal with low amplitude to the host data. This specific PN signal can later on be detected using a correlation receiver or matched filter. If the parameters like amplitude and the number of samples of the added PN signal are chosen appropriately, the probabilities of false-positive or false-negative detections are very low. The PN signal has the function of a secret key. The scheme can be extended if the PN signal is either added or subtracted from the host signal. In this case, the correlation receiver will calculate either a high-positive or high-negative correlation in the detection. Thus, 1 bit of information can be conveyed. If several such watermarks are embedded consecutively, arbitrary information can be conveyed.

## IV. TEXT DOCUMENT WATERMARKING

Methods for embedding information into text documents have been used for a long time by secret services.

For text watermarking, we have to distinguish between methods that hide information in the semantics, which means in the meaning and ordering of the words, and methods that hide information in the format, which means in the layout and the appearance.

The first class designs a text around the message to be hidden. In that sense, the information is not really embedded in existing information, but rather covered by misleading information. This class of techniques is outside the scope of this paper and will not be considered here. In the following, we concentrate on the latter type of information-embedding methods which use an existing text document into which data are embedded.

Formatted text is probably the medium where watermarking methods can be defeated most easily. If the watermark is in the format, then it can obviously be removed by "retyping" the whole text using a new character font and a new format where "retyping" can be either manual or automated using optical character recognition (OCR). OCR systems are still not perfect for many applications today and often need human supervision. Thus, removal of watermarks either yields bad results (single characters are wrong, due to OCR) or is expensive. The goal is to make watermark removal more expensive than obtaining the right to copy from the copyright owner. If this goal is achieved, text watermarking makes sense, though it can be defeated [14].

Text watermarking has applications wherever copyrighted electronic documents are distributed. Important examples are virtual digital libraries where users may download

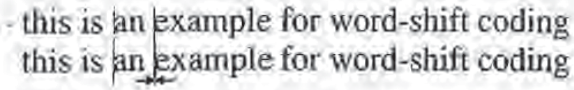


**Fig. 3.** Example for word-shift coding.

copies of documents, for example, books, but are not allowed to further distribute them or to store them longer than for a certain predefined period. In this type of application, a requested document is watermarked with a requester specific watermark before releasing it for download. If later on illegal copies are discovered, the embedded watermark can be used to determine the source.

Brassil *et al.* [14], [15], [84], [85], [91] have extensively worked on text watermarking. They propose three different methods for information embedding into text documents: line shift coding; word-shift coding; and feature coding. In line-shift coding, single lines of the document are shifted upwards or downwards by very small amounts. The information to be hidden is encoded in the way the lines are shifted. Similarly, words are shifted horizontally in order to modify the spaces between consecutive words in word-shift coding. An example for word-shift coding is shown in Fig. 3. Both methods are applicable to the format file of a document or to the bitmap of a page image. While line-shift coding can rely on the assumption that lines are uniformly spaced, and thus does not necessarily need the original for watermark extraction, the original is required for extraction in word-shift coding, since the spaces between words are usually variable. The third method, feature coding, slightly modifies features such as the length of the end lines in characters like $b, d, h$, etc. Among the three presented methods, line-shift coding is the most robust in the presence of noise but also most easily defeated. The authors again argue that although the described methods can theoretically be defeated, it requires interactive human intervention and is expensive in practice. The presented methods are robust enough to resist printing, consecutive photocopying up to ten generations, and rescanning [85].

## V. IMAGE WATERMARKING

Most watermarking research and publications are focused on images. The reason might be that there is a large demand for image watermarking products due to the fact that there are so many images available at no cost on the World Wide Web which need to be protected.

Meanwhile, the number of image watermarking publications is too large to give a complete survey over all proposed techniques. However, most techniques share common principles. Thus, we try to point out the common ideas first, before we explain some selected methods in more detail to illustrate how the principles are applied in practice.

The watermark signal is typically a pseudorandom signal with low amplitude, compared to the image amplitude, and usually with spatial distribution of one information (i.e., watermark) bit over many pixels. A lot of watermarking methods are in fact very similar and differ only in parts or single aspects of the three topics: signal design; embedding, and recovery.

The information that is embedded is usually not important for the watermarking itself. However, there are methods that are designed to embed and extract one out of a codebook of codes, and thus cannot accommodate arbitrary information [27], [72]. Other proposed schemes modulate the codes available in the codebook with arbitrary information bits and can thus accommodate arbitrary messages. Although some authors distinguish strictly between the two types, they are in fact conceptually very close.

The watermark signal is often designed as a white [136], [139] or colored pseudorandom signal with, e.g., Gaussian [27], uniform, or bipolar [33], [72], [76], [93], [136], probability density function (pdf). In order to avoid visibility of the embedded watermark, an implicit or explicit spatial [7], [66], [126], [146] or spectral [66], [105], [106], [136], [130], [146] shaping is often applied with the goal to attenuate the watermark in areas of the image where it would otherwise become visible. The resulting watermark signal is sometimes sparse and leaves image pixels unchanged [33], [74], but mostly it is dense and alters all pixels of the image to be watermarked. The watermark signal is often designed in the spatial domain, but sometimes also in a transform domain like the full-image discrete cosine transform (DCT) domain [27] or block-wise DCT domain [69].

The signal embedding is done by addition [78], [93], [139] or signal-adaptive (i.e., scaled) addition [2], mostly to the luminance channel alone, but sometimes also to color channels, or only to color channels [73]. The addition can take place in the spatial domain, or in transform domains such as the discrete Fourier transform (DFT) domain [113], the full-image DCT domain [3], [27], [105], the block-wise DCT domain [7], [47], [69], [78], [106], [151], the wavelet domain [71], [72], [143], the fractal domain [34], [96], [109], the Hadamard domain [59], [111], the Fourier–Mellin domain [114], [115], or the Radon domain [150]. It is often claimed that embedding in the transform (mostly DCT or wavelet) domain is advantageous in terms of visibility and security [3]. However, while some authors argue that the watermarks should be embedded into low frequencies [27], [114], other argue that they should rather be embedded into the medium [3], [36], [56] or high frequencies. In fact, it has been shown [122], [123] that for maximum robustness watermarks should be embedded signal adaptively into the same spectral components that the host data already populate. For images and video, these are typically the low frequencies.

As said before, watermark signal generation and watermark embedding are often treated jointly. For some proposed methods, they cannot be regarded separately, especially if the watermark is signal adaptive [3], [22], [23], [78], [148].

The watermark recovery is usually done by some sort of correlation method, like a correlation receiver or a matched filter. Since the watermark signal is often designed without knowledge of the host signal, crosstalk between watermark signal and host data is a common problem in watermarking. In order to suppress the crosstalk, many proposed schemes require the original, unwatermarked data in order to subtract it before watermark extraction. Other proposed methods apply a prefilter [38], [73], [82], [139] instead of subtracting the original. Yet other methods do not suppress the crosstalk [105]. Some researchers propose to use more sophisticated detectors than just simple correlation detectors, e.g., maximum *a-posteriori* (MAP) detectors [3]. Like for embedding, several domains have been proposed for watermark extraction, often corresponding to the domain that is proposed for embedding or for signal design. There are fewer publications where watermark embedding and extraction are proposed in different domains.

Before we look at some specific watermarking techniques in the different domains, we give a brief chronological overview of early watermarking methods.

The year 1993 can be considered the beginning of the digital image watermarking era, although other publications from the early 1990's, such as Tanaka *et al.* [131], [132], already introduced the idea of tagging images to secretly hide information and assure ownership rights. Caronni [20], [21] describes an overall system to track unauthorized image distribution. He proposes to mark images using spatial signal modulation and calls the process tagging. A tag is a square of size $N * N$. In a first step, all possible locations in an image where a tag could possibly be placed are identified by calculating the local region variance of size $N * N$ in the image and comparing it to empirically identified upper and lower limits. Only locations with minimal variance are used for tagging. A tag is a square with a constant value proportional to the maximum image brightness within the square and decaying outside the border. A selected image area is tagged by adding or subtracting the tag and a random, zero mean, noise pattern. Both the tag location and the noise sequence are key dependent. One selected tag location hides 1 bit and is only tagged if the bit to embed is set to one. To recover an embedded bit, the difference between the original and the tagged image is computed. Then the mean of a supposedly tagged location is compared to the neighboring mean to determine the bit value. In addition to the marking process, Caronni also suggests to use the correlation coefficient between the original and the tagged image as a measure for the image degradation due to the tagging process. A correlation coefficient of one indicates that the two images are identical, whereas for distorted images the value decreases toward zero.

In the same year, approaches and ideas for digital image watermarking were proposed by Tirkel *et al.* [136] in their 1993 publication entitled *Electronic Water Mark*. In this early publication on digital watermarking, the authors already recognized the importance of digital watermarking and proposed possible applications for image tagging, copyright enforcement, counterfeit protection, and controlled access to image data. Two methods were proposed for grayscale images. In the first approach, the watermark in form of an $m$-sequence-derived PN code is embedded in the least significant bit (LSB) plane of the image data. To gain full access to the LSB plane without introducing much distortion, the image is first compressed to 7 bits through adaptive histogram manipulation. This method is actually an extension to simple LSB coding schemes in which the LSB's are replaced by the coding information. The watermark decoding is straightforward since the LSB plane carries the watermark without any distortion. In the second approach, the watermark, again in form of an $m$-sequence-derived code, is added to the LSB plane. The decoding process makes use of the unique and optimal autocorrelation function of $m$-sequences [86]. A modified version of the paper was published in 1994 [139] titled *A Digital Watermark*, and being the first publication explicitly mentioning, and hence defining, the term digital watermarking. In 1995 [137], the idea of using $m$-sequences and LSB addition was extended and improved by the authors through the use of two-dimensional (2-D) $m$-sequences which resulted in more robust watermarks.

Table 1
Sample Cipher Key Table

| $\Delta_i$ | ⋯ | -4 | -3 | -2 | -1 | 0 | 1 | 2 | 3 | 4 | ⋯ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $c_i$ | ⋯ | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | ⋯ |

About the same time Matsui and Tanaka [90] published a paper called "Video Steganography: How to Secretly Embed a Signature in a Picture," in which several watermarking techniques were proposed for image watermarking. Their first method is based on a predictive coding scheme for gray scale images. Predictive coding schemes exploit the correlation between adjacent pixels by coding the prediction error instead of coding the individual gray scale values. A digital image is scanned in a predefined order traversing the pixels $\{x_i\}; i \in N$. The set of pixels is then coded using a predictive coding scheme by keeping the first value $x_1$ and replacing subsequent values $x_i$ by the difference $e_i$ between adjacent pixels

$$e_i = x_i - x_{i-1}. \tag{6}$$

To embed a watermark in form of a binary string, Matsui and Tanaka introduce a cipher key table which assigns a corresponding bit $c_i$ to all possible differences $\Delta_i$. An example of such a table is given in Table 1. The correspondence between bit values and the differences is kept secret. To embed a bit $b$, select a pixel $x_i$ with its corresponding difference $e_i$. Check in the cipher table if the bit value $c_i$ corresponding to $\Delta_i = e_i$ has the same value as bit $b$. If this is the case, proceed to the next bit, otherwise select the closest value to $e_i$ in the cipher table that has the appropriate bit value. The watermark can be recovered by looking up the bit in the coding table. The second method modifies the ordered dithering scheme for binary pictures. A dithering scheme consists of comparing the monotone level of pixels within a pixel block with a position-dependent threshold and turning "on" those pixels with a value above the threshold. The location dependent thresholds are given in a square matrix of size $N * N$ called dither matrix with entries $d_{p,q}^{(n)}$, where $n$ denotes an ordering number between zero and $N^2 - 1$ and $p$ and $q$ the

| | | | |
|---|---|---|---|
| 6 | 7 | 8 | 9 |
| 5 | 0 | 1 | 10 |
| 4 | 3 | 2 | 11 |
| 15 | 14 | 13 | 12 |

Fig. 4. Sample dither matrix: dot-concentrated type.

corresponding matrix line and column, respectively. Fig. 4 shows a sample dithering matrix. Given the dither matrix, the corresponding thresholds $T$ are defined as

$$T = \left(d_{pq}^{(n)} + \frac{1}{2}\right) \times \frac{R}{N^2} \tag{7}$$

where $R$ defines the dynamic brightness range of the image. To dither an image, it is first divided into adjacent blocks of the same size as the dither matrix. Then all values in each block are compared to the corresponding threshold value and modified accordingly. Now let the set set of threshold pairs be defined as

$$S_k = \{(x_i, x_j)_k | x_i - x_j = k; i, j = 0, 1, \ldots, N; i \neq j\} \tag{8}$$

where $x_{i,j}$ denote thresholds in the dither matrix. Further, let $(y_i, y_j)_k$ be the output signal of $x_i, x_j$ and assuming the values of $(0,0)_k$, $(0,1)_k$, $(1,0)_k$, and $(1,1)_k$. Only the two pairs (0, 1) and (1, 0) are considered for data embedding.

To embed a bit $b$, an output pair $(y_i, y_j)_k$ is selected, and $y_i$ is compared with the bit value $b$. If the values are equal, the pair is left unchanged, otherwise $y_i$ and $y_j$ are swapped. In order to decode an embedded signature, the above described procedure is inverted. Again, the pairs $(0,0)_k$ and $(1,1)_k$ are disregarded. The third scheme is proposed to watermark facsimile documents. Facsimile documents are scanned with a horizontal resolution of about 8.23 pixels/mm and then compressed using run length encoding (RLE) followed by modified Huffman coding (MH). The embedding process modifies the run lengths between two subsequent, changing pels. If a one is to be embedded, the run length is forced to be even, whereas for a zero the run length is forced to be odd. For valid embedding, the original run length has to be larger than one. Decoding an embedded bit is achieved by looking at the decoded run length. Their last method is based on the modification of DCT coefficients in a progressive transmission scheme. The watermark bits are embedded by modifying the rounding rule for the quantized coefficients such that the resulting coefficients are odd or even, depending on the watermark bits.

It was soon recognized that digital watermarking and digital modulation, and especially direct sequence spread-spectrum modulation [40], [102], [119], [140], share similar concepts, and it was proposed to consider digital watermarking as communication in non-Gaussian noise. First theoretical approaches were proposed by Smith [120]. A more in depth analysis of 2-D multipulse amplitude modulation was given by Hernandez *et al.* [53].

Since the above-mentioned first publications, the interest and research activities on watermarking have largely increased. Multimedia content providers and distributors are especially interested in working solutions. In the following, we present some of the more recent work and start the overview with methods working in the spatial domain.

Bender *et al.* [6] propose two methods for data hiding. In the first method, called "Patchwork," randomly selected pairs of pixels $(a_i, b_i)$ are used to hide 1 bit by increasing the $a_i$'s by one and decreasing the $b_i$'s by one. Provided that the image satisfies some statistical properties, the expected value of the sum of the differences between the $a_i$'s and $b_i$'s of $N$ pixel pairs is given by $2N$

$$\sum_N a_i - b_i = \begin{cases} 2N, & \text{for watermarked pairs} \\ 0, & \text{for nonwatermarked pairs.} \end{cases} \tag{9}$$

In the second approach, called "Texture Block Coding," the watermark is embedded by copying one image texture block to another area in the image with a similar texture. To recover the watermark, the autocorrelation function has to be computed. A remarkable feature of this technique is the high robustness to any kind of distortion, since both image areas are distorted in a similar way, which means that the watermark recovery by autocorrelation still works.

Pitas and Kaskalis propose signature casting on digital images [93], [103], [104], which is based on the same basic idea as the patchwork algorithm proposed by Bender *et al.* [6]. The watermark $S = \{s_{m,n}\}$ consists of a binary pattern of the same size as the original image and where the number of "ones" is equal to the number of "zeros." The original image $I$, with luminance values $x_{m,n}$ at location and $m$ and $n$, is divided into two sets $A$ and $B$ of equal size in the following way:

$$\begin{aligned} A &= \{x_{mn} \in I, s_{mn} = 1\} \\ B &= \{x_{mn} \in I, s_{mn} = 0\}. \end{aligned} \tag{10}$$

The watermark is superimposed by changing the elements of the subset $A$ by a positive integer factor $k$, e.g., $A' = \{x_{mn} + k, x_{mn} \in A\}$. The watermarked image is then given by the union of $A'$ and $B$. To verify the presence of a watermark, hypothesis testing [97] is applied. The test statistic $q$ is defined as the normalized difference between the mean $\bar{a}'$ of set $A'$ and the mean $\bar{b}$ of set $B$

$$q = \frac{\bar{b} - \bar{a}'}{\sigma_{A'}^2 + \sigma_B^2} \tag{11}$$

where $\sigma_{A'}^2$ and $\sigma_B^2$ defines the sample variance of set $A'$ and $B$, respectively. The test statistic is then compared with a threshold to determine if there is a watermark. The method is immune to subsampling followed by up-sampling and resists to JPEG compression with a compression factor of 1:4.

An improved version of this idea has been proposed Langelaar *et al.* [78], [82]. The image is tiled into square blocks with a size being a multiple of eight. A single bit is embedded by iteratively modifying a pseudorandomly selected block. Each selected block has a pseudorandom pattern $P$, with equal number of "1" and "0" assigned to it. To embed a bit with a value of "1," the scaled pattern $k \times P$, where $k$ is a predefined scaling factor defining the initial minimal watermark strength, is added to the block. For a bit with a value of "0," the scaled pattern is subtracted from the block. Let $I_0$ be the mean of all pixel values within the block for which the corresponding pattern value is zero, and $I_1$ the mean of the remaining pixels. Further, let $D_{\text{high}} = I_1 - I_0$ be the difference between the two means, and $D_{\text{low}} = \hat{I}_1 - \hat{I}_0$ be the difference between the means after JPEG compression of the block with a predefined quality factor $Q$. If a "0" is to be embedded, the pattern $P$ is iteratively subtracted from the block until both differences, $D_{\text{high}}$ and $D_{\text{low}}$ are below zero or the maximum number of iterations has been reached. If a "1" is to be embedded the pattern is iteratively added to the block until both differences, $D_{\text{high}}$ and $D_{\text{low}}$, are above a predefined threshold $T$ or the maximum number of iterations has been reached. An embedded bit can be extracted by again computing the difference $D_{\text{high}}$ between the two means $I_1$ and $I_0$. The sign of this difference is then used to determine the embedded bit value. Tests with the parameters set to block size 32 × 32, threshold $T = 1$, initial scaling factor $k = 4$ and maximum number of iterations six, indicate that the method features decent robustness toward JPEG compression with a bit error rate of about 5% for 85% JPEG quality and 20% for 60% JPEG quality. In a second method the authors propose watermarking in the DCT domain by setting DCT-coefficients below a selected scan line to zero.

To increase the performance of the block base spatial watermarking methods, Bruyndonckx *et al.* [17] suggest the used of pixel classification. Pixels within pseudorandomly selected blocks are classified into zones (1 and 2) of homogeneous luminance values. The classification is based on three types of contrast between zones: hard contrast; progressive contrast; and noise contrast. Each zone is then further subdivided into two categories $A$ and $B$ based on a grid defined by the coder. Each pixel is thus assigned to one of four zone/category combinations, e.g., $1/A, 1/B, 2/A$, and $2/B$. A bit $b$ is embedded by modifying the zone/category means to satisfy the following constraints:

$$\begin{aligned} \text{if } b = 0: \quad & m^*_{1B} - m^*_{1A} = S \\ & m^*_{2B} - m^*_{2A} = S \\ \text{if } b = 1: \quad & m^*_{1A} - m^*_{1B} = S \\ & m^*_{2A} - m^*_{2B} = S \end{aligned} \tag{12}$$

where $m^*_{1A}, m^*_{1B}, m^*_{2A}$, and $m^*_{2B}$ are the modified zone/category mean values and $S$ the watermark embedding strength. The modification of the mean values is done by applying equal luminance variations for all pixels belonging to the same zone. To increase robustness the authors suggest to perform redundant bit embedding and use error-correcting codes. Good robustness to JPEG compression is reported.

In order to increase the performance of spread-spectrum watermarking in the spatial domain Kutter *et al.* [73], [74] propose a method which exclusively works with the blue image component, in the RGB color space, in order to maximize the watermark strength while keeping visual artifacts minimal. Further, they propose to preprocess the image prior to watermark decoding in order to predict the embedded watermark. This concept improves the robustness significantly and is applicable to any spread-spectrum watermarking in the spatial domain. The method embeds a watermark in form of a binary number through amplitude modulation in the spatial domain. A single bit $b$ is embedded at a pseudorandomly selected location $(i, j)$ by either adding or subtracting, depending on the bit, a value which is proportional to the luminance at the same location

$$B_{i,j} \leftarrow B_{i,j} + \alpha(-1)^b L_{i,j} \tag{13}$$

where $B_{i,j}$ describes the blue value at location $(i,j)$, $L_{i,j}$, the luminance at the same location, and $\alpha$, the embedding strength. To recover an embedded bit, an estimate of the original, nonwatermarked, value is computed using linear combination of neighboring pixels in a cross shape

$$\hat{B}_{i,j} = \frac{1}{4c}\left(\sum_{k=-c}^{c} B_{i+k,j} + \sum_{k=-c}^{c} B_{i,j+k} - 2B_{i,j}\right) \tag{14}$$

where $c$ defines the size of the cross-shaped neighborhood. The bit value is determined by looking at the sign of the difference $\delta_{i,j}$ between the pixel under inspection and the estimated original. In order to increase robustness, each signature bit is embedded several times, and to extract the embedded bit the sign of the sum of all differences $\delta_{i,j}$ is used. Fig. 5 illustrates an image composition example. The two watermarked images on the top are used to generate the new composite image on the bottom. Given the appropriate keys, both original watermarks can be recovered. Extensions to this method allow increased robustness and even watermark recovery after geometrical attacks [76] and printing–scanning.

Macq *et al.* [37], [87] introduce watermarking adapted to the human visual system (HVS) using masking and modulation. In their scheme, the watermark in form of a spatially limited binary pattern is low-pass filtered, frequency modulated, masked, and then added to the host image. A secret key is used to determine the modulation frequencies and the watermark embedding location. The masking process uses an extension of the masking phenomena for monochromatic signals, also called gratings. To further adapt the watermark to the image, a shaping mask, based on morphological homogenized areas of high frequencies, is used. Watermark recovery is performed by demodulation followed by a correlation function.

In a very different approach, Voyatzis and Pitas watermark images by inserting logo like patterns using torus automorphisms [141], [142]. A 2-D torus automorphism can be considered as a spatial transformation of planar regions which belong to a square 2-D area. It is defined in the subset

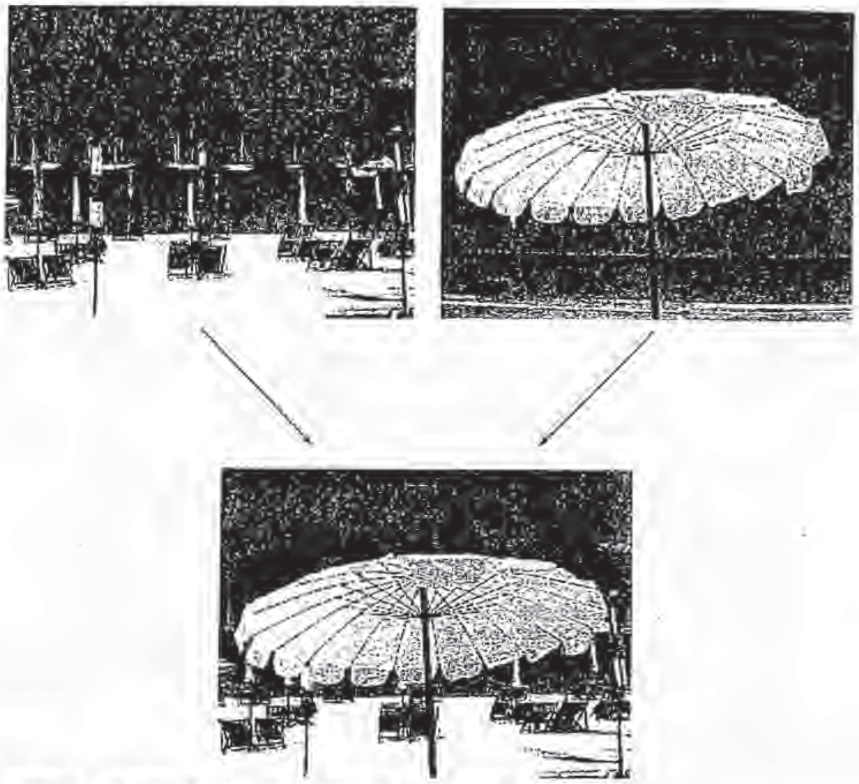


Fig. 5. Image composition. The umbrella of the "umbrella" image is pasted onto the "beach" image. The watermarks from both images can be recovered from the composed image.

$U = [0,1) \times [0,1) \subset R^2$ by

$$r' = Ar, \quad \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \pmod 1. \quad (15)$$

Iterated actions of $A$ on a point $\mathbf{r}_0$ form a dynamical system which can be expressed like a map

$$\mathbf{r}_{n+1} = A^n \mathbf{r}_0 \pmod 1 \quad \text{or} \quad \mathbf{r}_{n+1} = A\mathbf{r}_n \quad (16)$$

An example for a well-known automorphism in dynamics is the "cat map," defined as

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \pmod 1. \quad (17)$$

The set of points $\{\mathbf{r}_0, \mathbf{r}_1, \mathbf{r}_2, \cdots\}$ is called an orbit of the system. Roughly speaking, such a system mixes the points in a chaotic way. Under certain circumstances, the automorphisms may have periodic orbits, which means that after $T$ iterations the current point is equal to the initial point, e.g., $A^T \mathbf{r}_0 = \mathbf{r}_0$. Fig. 6 shows an example of an authomorphism using the cat map.

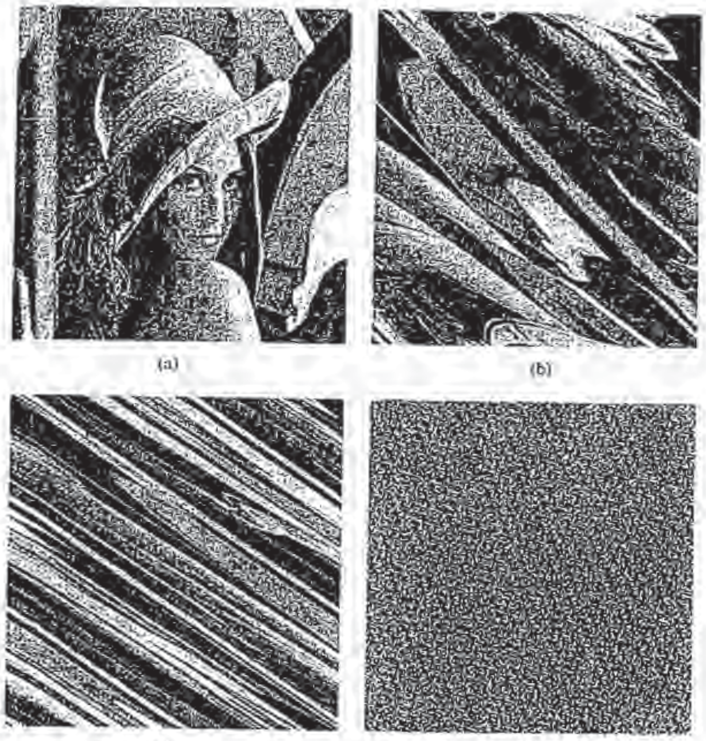
To sign an image, a watermark in the form of a square binary image, with a size smaller than the original image, is first mixed using the automorphism $A_N$. The resulting mixed watermark is then overlaid on a selected block in the original image using an embedding function such as LSB modification. Watermark recovery is performed by first extracting the mixed watermark from the signed image followed by reconstructing the watermark using the automorphism $A_{N-T}$, where $T$ is the automorphism period for the given system. Using more sophisticated overlaying methods will increase the robustness of the method.

Raymond and Wolfgang [147], [148] propose a watermarking technique to verify image authenticity based on an approach similar to the $m$-sequence approach suggested by Schyndel *et al.* for the one-dimensional case [139] and Tirkel *et al.* for the two dimensional case [137]. A random sequence generated by using linear feedback shift registers is mapped from {0,1} to {1, −1}, arranged into a suitable block and added to the image. To locate where an image has been forged, the algorithm overlays the watermarked image block with the watermark block, computes the inner product, and compares the result to the ideal value. Let the cross-correlation function $R_{XY}(\alpha, b\beta)$ of two blocks $X$ and $Y$ be defined as

$$R_{XY}(\alpha, b\beta) = \sum_i \sum_j X(i,j) Y(i-\alpha, j-\beta) \quad (18)$$

then the test statistic $\delta$ for a block, given the original image block $X$, the watermark block $W$, the watermarked image block $Y$, and the probably forged image block $Z$, is defined as

$$\delta = R_{YW}(0,0) - R_{ZW}(0,0). \quad (19)$$


(a) (b)

(c) (d)

Fig. 6. Example automorphism with the "cat map." (a) is the original image. (b)–(d) show the automorphism of (a) after one, two, and ten iterations, respectively.

If the watermark is unchanged, $\delta = 0$. When $\delta$ is greater than a defined tolerance, the block fails the watermark test. The method detects any kind of image filtering, and the authors claim that an improved version can even accommodate JPEG compression.

Watermark embedding not based on spread-spectrum modulation but quantization has been proposed by Chen and Wornell [24]. Their method is called quantized index modulation (QIM) and is based on a set of $N$-dimensional quantizers. The quantizers satisfy a distortion constraint and are designed such that the reconstruction values from one quantizer are "far away" from the reconstruction points of every other quantizer. The message to be transmitted is used as in index for quantizer selection. The selected quantizer is then used to embed the information by quantizing the image data in either the spatial or DCT domain. In the decoding process, a distance metric is evaluated for all quantizers and the index of the quantizer with the smallest distance identifies the embedded information. The authors show that the performance of the resulting watermarking scheme is superior to standard spread-spectrum modulation without watermark weighting.

Besides spatial domain watermarking related to modulation it was proposed by Maes *et al.* [89] to modify geometric features of the image. The method is based on a dense line pattern, generated pseudorandomly and representing the watermark. A set of salient points in the image is then computed, for example, based on an edge detection filter. The detected points are then warped such that a significantly large number of points are within the vicinity of lines. In the detection process, the method verifies if a significantly large number of points are within the vicinity of lines.

Related to spatial domain watermarking schemes are methods based on fractal image compression. The idea to use this approach has first been proposed by [109]. In fractal image compression the image is coded using the principles of iterated function systems and self similarity [116]. The original image is divided into square blocks $R_k$ called range blocks. Further, let $F$ be a set of mapping functions $f_k$, which are a composed of a geometric transformation $g_k$ and a massic transformation $m_k$. The mapping functions work on domain blocks $D_k$, which are larger than range blocks. The geometric transformation consists of moving the domain block $D_k$ to the location of the range block $R_k$ and reducing the size of the domain block to the size of the range block. The massic transformation adjusts the intensity and orientation of pixels in the domain block after geometric transformation. Massic transformations include rotation by 90, 180, and −90°, reflection at midhorizontal and cross-diagonal axis, as well as identity mapping. To compress an image for all range blocks $R_k$, the best combination of domain block $D_k$ and mapping function $f_k$ has to be found such that the difference between the range block $R_k$ and the mapped domain block $f_k(D_k)$ is minimal. That means that the encoding includes a spatial search over all possible domain blocks. Decoding is accomplished by iterating over the coded mapping functions using any initial image. To embed a bit into this scheme a range block is pseudorandomly selected. The corresponding search space $S_k$ for the range blocks is then split up into two subsearch spaces $S_k^1$ and $S_k^2$ of equal size. Each subspace is assigned to a bit value, and the current range block is encoded by searching only in the subspace corresponding to the bit value of the current bit. To recover an embedded bit, the image is again compressed, however this time using the full domain block search space. Then for a marked range block the location of the corresponding domain block reveals the embedded bit value. The algorithm was tested against JPEG compression and showed good robustness down to a compression quality of about 50%. A drawback of this technique is the slow speed due to the fractal compression scheme.

A very similar approach has been proposed by Davern and Scott [34]. The only difference is that they do not encode the entire image, but only a user-defined range region based on a user-defined domain region. Given the two regions, the watermark encoding is equivalent to the system proposed by Puate and Jordan in that the domain region is divided into two parts and, depending on the bit value, one or the other region is used for encoding a range block. This idea of watermarking using spatial domain fractal image coding has been extended to DCT blocks by Bas *et al.* [4].

Efficient watermarking in the DCT domain was first introduced by Koch *et al.* [18], [68], [69]. As in the JPEG compression scheme, the image is first divided into square blocks of size $8 \times 8$ for which the DCT is computed. From a pseudorandomly selected block, a pair of midfrequency coefficients is selected from 12 predetermined pairs. To embed a bit, the coefficients are then modified such that the difference between them is either positive or negative, depending on the bit value. In order to accommodate lossy JPEG compression, the quantization matrix is taken into account when altering the DCT coefficients. This method shows good robustness to JPEG compression down to a quality factor of 50%.

Bors and Pitas [12], [13] suggest a method that modifies DCT coefficients satisfying a block site selection constraint. The image is first divided into blocks of size $8 \times 8$. Certain blocks are then selected according to a Gaussian network classifier decision. The middle range frequency DCT coefficients are then modified, using either a linear DCT constraint or a circular DCT detection region, to convey the watermark information. In the first approach, the linear constraint is defined as

$$Y = FQ \tag{20}$$

where $F$ is the modified DCT coefficient vector and $Q$ the weighting vector provided by the watermark. The constraint is imposed by changing the DCT coefficients based on a least-squares criterion. The second algorithm defines circular regions around the selected DCT frequency coefficients. The selected frequencies are then quantized according to

$$\|F - Q_k\|^2 = \min_{l=1}^{H} \|F - Q_l\|^2 \quad \text{then} \quad F = Q_k \tag{21}$$

where $Q_l, l = 1, \ldots, H$ is the set of coefficient vectors provided by the watermark. In the watermark recovery process, the algorithm first verifies the DCT coefficient constraint for all blocks followed by the location constraint. The algorithm can accommodate JPEG compression for a compression ratio of 13:1 and 18:1 using the linear DCT constraint or the circular DCT detection region, respectively.

Swanson *et al.* [129], [130] suggest a DCT domain watermarking technique, based on frequency masking of DCT blocks, which is similar to methods proposed by Smith and Comiskey [120]. The input image is split up into square blocks for which the DCT is computed. For each DCT block, a frequency mask is computed based on the knowledge that a masking grating raises the visual threshold for signal gratings around the masking frequency. The resulting perceptual mask is scaled and multiplied by the DCT of a maximal length PN sequence. This watermark is then added to the corresponding DCT block followed by spatial masking to verify that the watermark is invisible and to control the scaling factor. Watermark detection requires the original image as well as the original watermark and is accomplished by hypothesis testing. The authors report good watermark robustness for JPEG compression, colored noise, and cropping.

Tao and Dickinson [133] introduce an adaptive DCT-domain watermarking technique based on a regional perceptual classifier with assigned sensitivity indexes. Th watermark is embedded in $N$ AC DCT coefficients. Th coefficients are selected as to have the smallest quantizatio step sizes according to the default JPEG compression tabl The selected coefficients $x_i$ are modified as follows:

$$\hat{x}_i = x_i + \max\left[x_i\alpha_m, \mathrm{sign}(x_i)\frac{D_i}{\kappa}\right] \tag{22}$$

where $\alpha_m$ defines the noise sensitivity index for the curre block, $D_i$ the quantization step for $X_i$, and $\kappa$ satisfie $5 \leq \kappa \leq 6$. It should be noted that the watermark sign is not generated randomly. Various approaches exist t determine the noise sensitivity by efficiently exploitin the masking effects of the HVS. The authors propose regional classification algorithm which classifies the bloc in one of six perceptual classes. The classification algorith exploits luminance masking, edge masking, and textur masking effects of the HVS. Namely the perceptual bloc classes from one to six are defined as: edge; uniform low sensitivity; moderately busy; busy; and very busy, i descending order of noise sensitivity. Each perceptual cla has a noise-sensitivity index assigned to it. Watermark r covery requires the original image as well as the waterma and is based on hypothesis testing. Experiments show th the method resists JPEG compression down to a qualit of 5% and can accommodate random noise with a peak signal-to-noise ratio (PSNR) of 22.1 dB.

Podilchuk [107], [108] introduces perceptual watermarking using the just noticeable difference (JND) to determine an image-dependent watermark modulation mask. The watermark modulation onto selected coefficients in either the DCT or wavelet transform domain can be described as

$$I^*_{u,v} = \begin{cases} I_{u,v} + \mathrm{JND}_{u,v} \times w_{u,v}, & \text{if } I_{u,v} > \mathrm{JND}_{u,v} \\ I_{u,v}, & \text{otherwise} \end{cases} \tag{23}$$

where $I_{u,v}$ are the transform coefficients of the original image, $w_{u,v}$ are the watermark values, and $\mathrm{JND}_{u,v}$ is the computed JND based on visual models. For DCT coefficients, the author suggest using a perceptual model defined by Watson based on utilizing frequency and brightness sensitivity as well as local contrast masking. This model provides image-dependent masking thresholds for each 8 × 8 DCT block. Watermark detection is based on the correlation between the difference of the original image and the image under inspection and the watermark sequence. The maximum correlation is compared to a threshold to determine whether an image contains the watermark in question. Experiments showed that the watermark scheme is extremely robust to JPEG compression, cropping, scaling, additive noise, gamma correction, and printing–xeroxing–scanning. For attacks involving a geometrical transformation, the inverse operation has to be applied to the image before the watermark-detection process.

Piva *et al.* describe another DCT-based method which exploits the masking characteristics of the HVS [105]. The watermark consists of a pseudorandom sequence of $M$ real numbers with normal distribution $\boldsymbol{X} = \{x_1, x_2, \ldots, x_M\}$. The coefficients of the $N \times N$ DCT of the original image $I$ are reordered into a vector using a zig-zag scan. From this vector, $M$ coefficients, starting at position $L+1$, are selected to generate the vector $\boldsymbol{T} = \{t_1, t_2, \ldots, t_M\}$. The watermark $\boldsymbol{X}$ is embedded into $\boldsymbol{T}$ according to

$$t'_i = t_i + \alpha|t_i|x_i \tag{24}$$

where $\alpha$ determines the watermark strength. The modified coefficients replace the nonmodified coefficients before the watermarked image $I'$ is reconstructed. In order to enhance the robustness visual masking is applied as follows:

$$y''_{ij} = y_{ij}(1-\beta_{ij}) + \beta_{ij}y'_{ij} = y_{ij} + \beta_{ij}(y'_{ij} - y_{ij}) \tag{25}$$

where $\beta_{ij}$ is a weighting factor taking into account the characteristics of the HVS. A simple way of choosing $\beta_{ij}$ is the normalized sample variance at pixel $y_{ij}$ defined as the ratio between the sample variance for a square block with center at $y_{ij}$ and the maximum of all block variances. As in most schemes, watermark detection is performed by comparing the correlation $z$ between the watermark and the possibly corrupted signed DCT coefficients $\boldsymbol{T}^*$ with a threshold $S_z$. The correlation $z$ is defined as

$$z = \frac{\boldsymbol{X} \cdot \boldsymbol{T}^*}{M} = \frac{1}{M}\sum_{i=1}^{M} x_i t_i^*. \tag{26}$$

The threshold $S_z$ is adaptive and given as

$$S_z = \frac{\alpha}{3M}\sum_{i=1}^{M} |t_i^*|. \tag{27}$$

Experimental results demonstrate that the watermark is robust to several image processing techniques (for example, JPEG compression, median filtering, and multiple watermarking) and geometrical distortions (after applying the inverse geometric transformation).

Frequency-domain watermarking was first introduced by Boland *et al.* [8] and Cox *et al.* [27], who independently developed perceptually adaptive methods based on modulation. Cox *et al.* draw parallels between their technology and spread-spectrum communication since the watermark is spread over a set of visually important frequency components. The watermark consists of a sequence of numbers $x = x_1, \ldots, x_n$ with a given statistical distribution, such as a normal distribution $N(0,1)$ with zero mean and a variance of one. The watermark is inserted into the image $V$ to produce the watermarked image $V'$. Three techniques are proposed for watermark insertion

$$v'_i = v_i + \alpha x_i \tag{28}$$

$$v'_i = v_i(1 + \alpha x_i) \tag{29}$$

$$v'_i = v_i e^{\alpha x_i} \tag{30}$$

where $\alpha$ determines the watermark strength and the $v_i$'s are perceptually significant spectral components. Equation (28) is only suitable if the values $v_i$ do not vary too much. Equations (29) and (30) give similar results for small values of $\alpha x_i$, and for positive $v_i$'s (30) may even be viewed as an application of (28) where the logarithms of the original values are used. In most cases (29) is used. The scheme can be generalized by introducing multiple scaling parameters $\alpha_i$ as to adapt to the different spectral components and thus reduce visual artifacts. To verify the presence of the watermark, the similarity between the recovered watermark $X^*$, given by the difference between the original image $V$ and the possibly tampered image $V^*$, and the original watermark $X$ is measured. The similarity measure is given by the normalized correlation coefficient

$$\mathrm{sim}(X, X^*) = \frac{X^* \cdot X}{\sqrt{X^* \cdot X^*}}. \tag{31}$$

Robustness tests showed that the method resists JPEG compression (at a quality factor of 5% and no smoothing), dithering, fax transmission, printing–photocopying–scanning, multiple watermarking, and collusion attacks. For the experiments, the watermark was of length 1000 with $N(0,1)$ [where $N(\mu,\sigma)$ represents a normal distribution with mean $\mu$ and variance $\sigma$], $\alpha$ was set to 0.1, and the watermark was embedded into the 1000 strongest DCT coefficients using (29). Boland *et al.* propose a similar technique based on a hybrid between amplitude modulation and frequency shift keying and suggest the use of different transform domains such as DCT, wavelet transform, Walsh–Hadamard transform, and the fast Fourier transform (FFT).

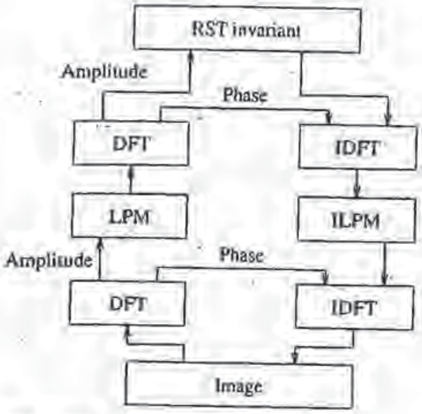


Fig. 7. RST invariant watermarking scheme.

Ruanaidh *et al.* propose watermarking by modification of the phase in the frequency domain [112], [113]. To embed a bit the phase of a selected coefficient $F(k_1, k_2)$ of an $N_1$ by $N_2$, DFT is modified by adding a small $\delta$

$$\angle F(k_1, k_2) \leftarrow \angle F(k_1, k_2) + \delta \tag{32}$$

in order for the watermarked image to be real, the phase must satisfy negative symmetry, which leads to the additional modification

$$\angle F(N_1 - k_1, N_2 k_2) \leftarrow \angle F(N_1 - k_1, N_2 - k_2) + \delta. \tag{33}$$

Coefficients are only modified if their relative power is above a given threshold. If the original image is available, the watermark can easily be recovered by comparing the phase. In case the original is not available, Ruanaidh suggests prequantizing the original phase prior to modifying it. Then deviations between the quantized states could be used to convey the data.

In another publication, Ruanaidh *et al.* explicitly design a watermarking technique invariant to translation, rotation, and scaling [114]. The method is a hybrid between DFT and log-polar mapping. The process is depicted in Fig. 7. In a first step, the DFT of the image is computed. One of the DFT properties is that a shift in the spatial domain results in a phase shift in the frequency domain. Keeping only the amplitude for further processing makes the image translation invariant. In the second step, rotation and scale invariance is achieved by mapping the amplitude from the Cartesian grid to a log-polar grid. Consider a point $(x, y) \in \Re$, then the mapping is defined as

$$\begin{aligned} x &= e^{\mu} \cos\theta \\ y &= e^{\mu} \sin\theta \end{aligned} \tag{34}$$

where $\mu \in \Re$ and $0 < \theta 2\pi$. One can easily see that this is a one-to-one mapping and that rotation and scaling in the Cartesian grid are converted to a translation of the $\mu$ and $\theta$ coordinates, respectively. Computing again the DFT of the log-polar map and keeping only the amplitude results in a rotation and translation invariant. Taking the Fourier transform of a log-polar map is equivalent to computing the Fourier–Mellin transform. Hence combining the two steps results in a rotation, scale, and translation (RST) transformation invariant. The watermark takes the form of a two dimensional spread-spectrum signal in the RST transformation invariant domain. In a test, a 104-bit watermark was embedded into an image. The watermarked image was then rotated by 143 ° and scaled by 75% along each axis. The embedded watermark was recovered from this image. Further, the method resists JPEG compression down to a quality factor of 75% and cropping to 50% of the original image size. This approach, which is actually the first one which was especially designed as to resist to geometrical attacks, has interesting aspects and ideas and might trigger a new way of approaching the design of future watermarking techniques. A variation of this idea based on the Radon transform has been proposed by Wu *et al.* [150].

Embedding the watermark using a multiresolution decomposition has first been proposed by Boland *et al.* [8]. As for schemes working in other transformation domains, the watermark is usually given by a pseudorandom 2-D pattern. Both the image and watermark are decomposed using a 2-D wavelet transform, and in each subband of the image a weighted version of the watermark is added. Watermark decoding is, as usual, based on a normalized correlation between and estimate of the embedded watermark and the watermark itself. Various wavelet based schemes have been proposed [58], [71], [151], [152]. The difference between the schemes usually lies in the way the watermark is weighted in order to decrease visual artifact.

In this section we have presented several different watermarking methods. It can be recognized that most watermarking methods are based on the same basic principle: small, pseudorandom changes are applied to selected coefficients in the spatial or transform domain. This changes are later on identified by correlation or correlation-like similarity measures. Usually, the number of modified coefficients is much larger than the number of information bits to be encoded. This can be considered as redundant embedding and leads to implicit robustness. As we have seen, the watermark embedding domain may have a substantial influence on the watermark robustness. Spatial domain watermarking schemes are in general less robust toward noise like attacks, for example, due to lossy JPEG compression. However, a big advantage is the fact that the watermark may easily be recovered if the image has been cropped or translated. This is less obvious if the frequency domain is used. Cropping in the spatial domain results in a substantially large distortion in the frequency domain, which usually destroys the embedded watermark. The same is true for the full-frame DCT domain. If DCT blocks are watermarked, it is important to know the block position for successful watermark decoding. The wavelet domain has very similar drawbacks because the wavelet transform is neither shift nor rotation invariant. Most proposed methods watermark in the spatial domain. This is probably due to the simplicity and efficiency of such methods. The number of publications on DCT-based methods is also large.

## VI. VIDEO WATERMARKING

Video sequences consist of a series of consecutive and equally time-spaced still images. Thus, the general problem of watermarking seems very similar for images and video sequences, and the idea that image watermarking techniques are directly applicable to video sequences is obvious. This is partly true, and there are a lot of publications on image watermarking which conclude with the remark that the proposed approach is also applicable to video. Since image watermarking has been covered in great detail in Section V, we do not repeat it here, even if some of them carry the word video in the title [26]. However, there are also some important differences between images and video which suggest specific approaches for video.

One important difference is the available signal space. For images, the signal space is very limited. This motivates many researchers to employ implicit or explicit models of the HVS, in order to reach the threshold of visibility and to embed a watermark as robust as possible without sacrificing image quality. Examples have been cited in Section V. For video, the available signal space, i.e., the number of pixels, is much larger. On the other hand, video watermarking often imposes real-time or near-real-time constraints on the watermarking system. As a consequence, it is less important, and for many applications even prohibitively complex, to use watermarking methods based on explicit models of the HVS. Complexity in general is a much more important issue for video watermarking applications than it is for image watermarking applications.

For individual watermarking, i.e., fingerprinting, of video sequences (for example, embedding of a receiver ID), this problem is even more severe because video sequences are usually stored in compressed format. Uncompressed storage and on-the-fly compression, or decompression, watermarking, and recompression, are usually not feasible for this kind of application, unlike for images. Thus, such applications may require compressed-domain watermarking, as presented in [47], [49], and [80] and discussed below.

Another point to consider is that the structure of video as a sequence of still images gives rise to particular attacks, for example, frame averaging, frame dropping, and frame swapping [47], [126]. At frame rates of 25–30 Hz, as they are used in television, this would possibly not be perceived by the casual viewer. A good watermarking scheme, however, should be able to resist to this kind of attack, for example, by distributing watermark information over several consecutive frames. On the other hand, it might be desirable to retrieve the full watermark information from a short part of the sequence. It depends on the application of which of those two competing requirements is realized (or both, e.g., by embedding a multiscale watermark with more than one temporal scale [126] or progressive watermark transmission [33]).

While a lot of research has been published on image watermarking, there are fewer publications that deal with video watermarking. However, the interest in such techniques is high, for example, the emerging digital versatile disk (DVD) standard which will contain a copy protection system employing watermarking.[1] The goal is to mark all copyrighted video material such that DVD standard compliant players or recorders will refuse to play back or record pirated material.

In the following, some watermarking methods exploiting uncompressed or compressed video properties are discussed. Some other methods that have been proposed but are in fact image watermarking techniques applied to image sequences with or without subsequent compression are not discussed here.

Hartung and Girod [47]–[49] have concentrated on watermarking of compressed video for fingerprinting applications. They employ a straightforward spread-spectrum approach and embed an additive watermark into the video. The watermark is generated using a PN signal with the same dimensions as the video signal that is modulated with the information bits to be conveyed. Each information bit is redundantly embedded into many pixels. For each compressed video frame, the corresponding watermark signal frame is DCT transformed on an 8 × 8 block-by-block basis, and the resulting DCT coefficients are added to the DCT coefficients of the video as encoded in the video bitstream. This is done for $I$, $P$, and $B$ frames. A rate control is realized by individually comparing the number of bits for each encoded watermarked DCT coefficient versus the corresponding encoded unwatermarked coefficient. Due to variable length coding, the watermarked coefficient may or may not need more bits for encoding than the unwatermarked one. If more bits are required, and the bit rate of the video sequence may not be increased, the coefficient is not used for embedding. Due to the inherent redundancy in the watermark, the watermark information can still be conveyed as long as enough coefficients can be embedded. Visible artifacts, as they could be produced due to the iterative structure of hybrid video coding, are avoided by applying a drift compensation scheme. The added drift compensation signal is the difference of the motion compensated predictions from the unwatermarked and the watermarked sequence. Fig. 8 shows a basic block diagram of the method. The bit stream has to be parsed and the watermark has to be transformed with the DCT. However, the method does not require full decompression and recompression. The complexity of the scheme is in the same order of magnitude as decompression, and the embedded watermarks pertain in the video after decompression. The scheme is compatible with all DCT-based hybrid compression schemes, for example, MPEG-2, MPEG-4, and ITU-T H.263. MPEG-4 has tools for compression of arbitrarily shaped objects. For nonrectangular border blocks of such objects, the shape-adaptive DCT (SA-DCT) [118] is used instead of the DCT. The watermarking scheme is also applicable to such border blocks, only that the DCT of the watermark has to be replaced by the SA-DCT. The watermark is recovered from the decompressed video by correlation using the same PN sequence that was used

---

[1] As of April 1999, two competing proposals from two different industry consortia are under evaluation.

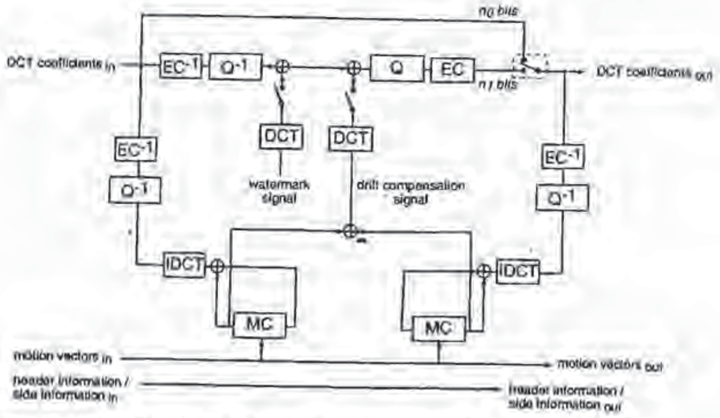


Fig. 8. Block diagram of watermark embedding into DCT coefficients of compressed video.

for generation of the embedded watermark signal. Typical watermark data rates are up to 50 bits/s, depending on the robustness requirements. The watermarks are robust against standard signal processing and with a modified watermark detector, as proposed in [50] also, to a certain extent, against geometrical distortions like shift, zoom, and rotation.

Jordan *et al.* [62] have proposed a method for the watermarking of compressed video that embeds information in the motion vectors of motion-compensated prediction schemes. Motion vectors pointing to flat areas are slightly modified in a pseudorandom way. Because the blocks pointed to by the original and the modified vectors are very similar (there is not much detail), this does not introduce any visible artifacts. The embedded information can be retrieved directly from the motion vectors, as long as the video is in compressed format. After decompression, the watermark can still be retrieved by first recompressing the video. This works because during recompression the watermarked motion vectors will be found with a probability high enough to statistically recover the watermark. The complexity of the method is negligible.

Hsu and Wu present a watermarking method [56], [57] for compressed video which is an extension of their method for images [55] and which modifies middle-frequency DCT coefficients in relation to spatially (for I-frames) or temporally (for P- and B-blocks) neighboring blocks. The coefficients are forced to assume a smaller or larger value than the corresponding neighboring coefficients, depending on the watermark sample to be embedded into the specific coefficient. The watermark signal is a visual pattern, like a logo, consisting of binary pixels. Prior to embedding, the watermark signal is spatially scrambled such that it can be recovered from a cropped version of the video. A drawback of the scheme is that for watermark extraction the watermarked video, the unwatermarked video, and the watermark have to be known.

In [80], Langelaar *et al.* propose two different information embedding schemes for compressed video. According to the different robustness and the definitions that we made in Section II, we call one of the methods a data-hiding method and the other a watermarking method. The data-hiding method adds the label directly in the MPEG-1 or MPEG-2 bit stream by replacing variable length codes (VLC'S) of DCT coefficients. In MPEG (and other hybrid coding schemes), the quantized DCT coefficients are encoded using run/level encoding and subsequent variable length coding. In the MPEG-2 code tables there exist pairs of codes which represent the same run and levels that deviate only by one from each other. One of the codes is then assigned a "1," the other one a "0." The idea is to find VLC's in the bit stream for which such a "similar" code exists, and to eventually replace one by the other such that the bit to be embedded is coded in the choice of the VLC. In principle, this could be done for intra- and intercoded blocks, but the authors alter only intracoded blocks. Still, they can embed up to 8 kbits/s into TV resolution video. The authors do admit, however, that the label can be removed easily by decompression and recompression without seriously affecting the video quality. The watermarking method is more complex, but also more robust. It is based on discarding parts of the compressed video bitstream. For each information bit to be embedded, a set of $n$ $8 \times 8$-blocks is pseudorandomly taken from the video frame and, also pseudorandomly, divided into two subsets of equal size. $n$ typically varies between 16 and 64. For each of the two subsets, the energy of the high-frequency DCT coefficients is measured. In order to embed the bit, the energy of the high-frequency coefficients in one or the other subset is reduced by removing high-frequency coefficients. The principle is illustrated in Fig. 9. For ease of understanding, consecutive blocks are used, rather than blocks randomly taken from the image. The information bit can be extracted by selecting the same set of blocks, dividing it into the same subsets, and comparing the energy of the high-frequency coefficients in each of the two subsets. Thus, the selection of blocks is the secret key involved. The method requires only partial decoding and no re-encoding. For TV resolution, up to 400 bits/s can be embedded. However, the robustness is limited. Re-encoding increases the error rate of the embedded bits much, and the method does not resist re-encoding using another group-

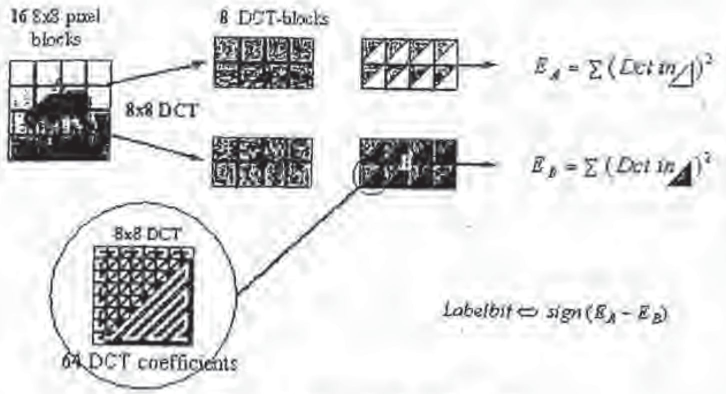


Fig. 9. Principle of DCT watermarking by comparison of the energy in the high-frequent coefficients. (Courtesy of G. Langelaar.)

of-picture (GOP) structure, since the DCT coefficients of a block are different depending on whether the frame is encoded as I, P, or B frame (however, in this case it is possible to extract the watermark by decoding and re-encoding the sequence with the same GOP structure that it had during watermarking [77]). Since DCT coefficients of the video are removed, care must be taken to adjust the parameters properly [79] in order to avoid visible blurring.

Swanson *et al.* [126], [127] propose a multiscale watermarking method working on uncompressed video which has some interesting properties. In a first step, the video sequence to be watermarked is segmented into scenes. Each scene is handled as an entity in the following. A temporal wavelet transform is then applied to each video scene, yielding temporal low-pass and high-pass frames. The watermark to be embedded is not an arbitrary message, but rather a unique code identifying the IPR owner and taken from a predefined codebook. In the design of the watermark, an explicit model of the HVS is employed in order to exploit spatial and temporal masking. Also, the watermark is designed with a signal-dependent key and thus avoids deadlock problems, as addressed in [30]. The watermark is embedded into each of the temporal components of the temporal wavelet transform, and the watermarked coefficients are then inversely transformed to get the watermarked video. Thus, the watermark has some components that change over time, while others do not or only slowly change over time, since they are embedded in the coefficients representing low temporal frequencies. This allows robustness against attacks like frame averaging, frame dropping, and the detection of the watermark from a frame of the scene without knowledge of its actual index. This is a property that the other video watermarking methods mentioned here do not automatically have. (Other video watermarking schemes could, however, achieve that with appropriate design of the watermark that they embed.) The watermark detection is done by hypothesis testing (the watermark is there or the watermark is not there). Experimental results show the robustness of the scheme against additive noise, MPEG video compression, and even frame drop. A disadvantage of the scheme is that it has a very high complexity, since it involves a forward and a backward wavelet transform, and an explicit model of the HVS including a blockwise DCT.

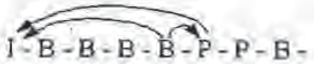


Fig. 10. Example for the structure of I, P, and B frames in a GOP.

Linnartz *et al.* [83] propose to embed information encoded in the GOP structure of the MPEG-2 compressed video. In MPEG-2, video frames can be encoded in three different ways: as intracoded I frames coded JPEG like and without reference to other frames; as P frames predicted from previous frames; or as B frames bidirectionally predicted from previous and following frames. I frames are needed as random access points. Usually, there is a maximum distance between two successive I frames in order to allow random access with a maximum delay. The frame type is signaled in the frame header and can be switched randomly from frame to frame. The set of frames from one I frame (including the I frame) to the next (excluding the next) is referred to as GOP (see Fig. 10). Possible GOP structures are for example "IPPP," "IBBPBBPBBPBB," "IBBBBBBBB," or "IPBPBBB," and in fact there are $2^{N-1}$ possible GOP structures for GOP's of N frames. A popular GOP size is, for example, $N = 12$, thus allowing as many as 2048 different variations. However, most available video codecs use a fixed GOP size and structure, and never use most of the admissible GOP structures. The idea for data embedding is to purposely use those (irregular) GOP structures, that are very unlikely, to embed information. Linnartz *et al.* propose a scheme where they embed 6 bits of information per GOP, which means very few bytes per second. The method can only be employed during compression, not after compression where the GOP structure is already fixed. Also, information embedded as such is not resistant to decompression. Thus, decompression and recompression would already remove this information completely. Another disadvantage might be that this type of watermark contradicts efforts to improve coding efficiency using rate-distortion optimized rate control [145], because such rate-distortion optimized video codecs are not restricted to a predefined GOP structure. A plus of the method is certainly that its complexity is negligible.

Darmstaedter *et al.* [33] propose to embed a spatial-domain low-pass spread-spectrum watermark into 8 × 8 pixel blocks of video sequences. The blocks are first classified according to their activity. Blocks with low activity are not watermarked. A low-pass pseudorandom pattern is then added to each selected block. In principle, each block (64 pixels) conveys one bit watermark information, but the bits are redundantly repeated over several blocks and several frames. Also, the authors apply an error correcting code. After watermark embedding, the sequence is compressed using MPEG-2 compression. Watermark extraction is done in the spatial domain after decompression using a correlation concept with thresholding. In order to achieve error-free watermark retrieval for compression down to a video bit rate of 6 Mbit/s, the authors embed one bit of watermark information into a total of 162 000 pixels.[2] The authors have verified the method, including real transmission over digital satellite links, and optimized the embedding parameters manually. Depending on block mean and block variance, the individual pixels (PCM encoded with 8 bit) are modified by up to ±6.

Dittmann *et al.* [39] apply two previously proposed still image watermarking methods [44], [69] to video. The video is decompressed prior to watermarking and recompressed after watermarking. The authors are not precise about video formats, encoding parameters, or other details, but they admit that after recompression, and using an error correcting BCH (31, 6, 15) code, residual bit error rates of 1–5% for the watermark information bits remain. Already with slight attacks like format conversion from MPEG-2 to Quicktime, the bit error rates increase significantly. Thus, at least the parameters of the scheme are obviously not chosen adequately.

Deguillaume *et al.* [36] propose to embed a spread-spectrum watermark into 3-D blocks of video by employing a 3-D DFT and adding to the transform coefficients. The watermark is composed of the real watermark and an auxiliary pattern, called template, that is easy to detect even under geometric attacks and that can be used to undo such attacks to enable retrieval of the real watermark. The blocks that are processed consist of typically 16 or 32 frames. Since the template is embedded into the 3-D log-log-log map of the DFT, it is not affected by zoom and shift [115]. Results are reported for an 88-bit watermark embedded into 3-D blocks of 32 CIF resolution (352 × 288 pixels) frames each (giving a watermark data rate of 1 bit per 36 864 pixels). The reported bit error rates are 0% after high-quality compression (bit rate 4.75 Mbits/s for CIF 25 Hz [35]), but without attack, and they go up to around 20% in the presence of aspect-ratio changes and frame-rate changes, even though the changes are recognized with help of the template and compensated. Thus, it seems that the parameters of the scheme should be chosen such that the watermark is embedded more robustly than presented in the simulations.

Busch *et al.* [19] apply a still-image watermarking method working on DCT blocks [69] to video sequences. The watermarks are embedded into the luminance component of uncompressed video and retrieved after decompression. In order to improve the invisibility of the watermarks, especially at edges, blocks are selected for watermarking depending on the block activity. For watermarking and watermark retrieval of a 64-bit watermark into each frame of ITU-R 601 video (that means into 5280 pixels/bit) and subsequent MPEG-2 compression at 4–6 Mbit/s, bit error rates between ≈0 and 50% are reported, depending on the sequence. For critical sequences, the authors propose to introduce additional temporal redundancy by embedding the watermark into several consecutive frames and averaging in the retrieval. For individual difficult sequences, averaging over 50 frames (corresponding to the embedding of one watermark bit into 264 000 pixels) still yields bit error rates of a few percent, and the authors propose averaging over an even higher number of frames for synthetic video.

Kalker *et al.* [65] have developed a video watermarking method for video broadcast monitoring applications which they call JAWS (just another watermarking system). For the sake of low complexity, both watermark embedding and detection are performed in the spatial domain, which means prior to compression and after decompression, respectively. The embedded watermark consists of watermark patterns of size 128 × 128 drawn from a white random process with Gaussian distribution that are repeated (tiled) to fill the whole video frame. In order to avoid visible artifacts, the watermark is, on a pixel-by-pixel basis, scaled with a scaling factor which is derived from an activity measure. The activity measure is computed using a Laplacian high-pass filter. The same watermark is embedded into several consecutive video frames. For watermark detection, a correlation detector is used after applying a spatial prefilter that reduces cross talk between video signal and watermark. Since the watermark must be detected even in the presence of spatial shifts, a search over all possible shifts is performed. Since the watermark signal is generated by tiling of a smaller watermark pattern, only 128 × 128 positions have to be searched, according to the size of the watermark pattern. In order to reduce complexity, the search and correlation is done in the FFT domain. Further, only the phase information of the FFT is used in the correlation. This method of detection has been previously proposed for pattern recognition and is referred to as symmetrical phase only filtering (SPOMF). In order to embed arbitrary watermark information, the watermark signal is designed using several different basic watermark patterns. The information is encoded in the choice of the basic patterns and their relative positions. The watermark can convey up to about 35–50 bits/s, but for applications

---

[2] 64 bits are embedded into 25 frames of ITU-R 601 resolution video (720 × 576 pixels).

that require less watermark information per second the watermark data rate is reduced for increased robustness [63]. The methods is claimed to be robust against MPEG-2 compression down to 2 Mbits/s, format conversion, scaling, and addition of noise.

Summarizing the above mentioned watermarking methods for video, a few general observations can be made.

1) The proposed methods span a wide complexity range from very low complexity to considerable complexity including, e.g., wavelet transforms and models of the HVS. In general however, the more complex methods seem to embed the watermarks with higher robustness.
2) Most methods operate on uncompressed video; only a few methods can embed watermarks directly into compressed video. For watermarking of compressed video watermarks can be embedded in the DCT coefficients [47], [49], [80], in the motion vectors [62], or in side information like the GOP structure [83].
3) The reported watermark data rates are between a few hundred bits per second and a few bits per second for television resolution video. It seems that if robustness is a real concern realistic data watermark data rates are not higher than a few bits per second to a few dozen bits per second. However, this is sufficient for most applications, including DVD.

## VII. AUDIO WATERMARKING

Compared to images and video, audio signals are represented by much less samples per time interval. This alone indicates that the amount of information that can be embedded robustly and inaudibly is much lower than for visual media. An additional problem in audio watermarking is that the human audible system (HAS) is much more sensitive than the HVS, and that inaudibility is much more difficult to achieve than invisibility for images.

Boney *et al.* [11] propose a spread-spectrum approach for audio watermarking. They use a PN sequence that is filtered in several stages in order to exploit long-term and short-term masking effects of the HAS. In order to exploit long-term masking, a masking threshold for each overlapping block of 512 samples is calculated and approximated using a tenth-order all-pole filter which is then applied on the PN sequence. Short-term masking is additionally exploited by weighting the filtered PN sequence with the relative time-varying energy of the signal in order to attenuate the watermark signal where the audio signal energy is low. Additionally, the watermark is low-pass filtered by using a full audio compression/decompression scheme as low pass, in order to guarantee that it survives audio compression. A high-pass component of the watermark is also embedded which improves watermark detection from uncompressed audio pieces but is expected to be removed by compression. The authors denote the two spectral components of the watermark by "low-frequency watermark" and "coding error watermark." The watermark can be extracted by hypothesis testing using the original and the PN sequence and by employing a correlation method. Experimental results show the robustness of the scheme to MPEG-1 layer III audio coding, to coarse PCM quantization using word lengths down to 6 bits/sample instead of 16 bits/sample as for the original, and additive noise.

Bassia and Pitas [5] apply a very straightforward time-domain spread-spectrum watermarking method to audio signals. They report robustness against audio compression, filtering and resampling.

Tilki and Beex [134] have developed a system for an interactive television application where they embed information into the audio component of a television signal. The embedded information is detected from the acoustic signal emitted from the television receiver. Though the system is designed for analog transmission, the principle could similarly be applied to digital signals. The information to be embedded is partitioned in blocks of 35 bits. Each information bit is modulated using a sinusoidal carrier of a specific frequency and low amplitude and added to the audio signal. The simplified principle is that if the sinusoidal carrier for a specific bit is present in the signal, the bit is "1," otherwise it is "0." The frequencies of the sinusoidal carriers are above 2.4 kHz, thus at frequencies where the HAS is less sensitive, no explicit model of the HAS is employed. In order to reduce interference from the audio signal itself, the audio signal is attenuated at frequencies above 2.4 kHz. Thus, the principle involves a fidelity loss of the host signal which seems acceptable for the envisaged application. In order to increase the robustness, the information bits are protected by a cyclic redundancy code (CRC) and bit repetition. In order to compensate frequency shift of the whole signal, for example, after analog recording and playback with inaccurate speed, a frequency locking mechanism is applied using five special sinusoidal carriers of known frequency. Thus, the scheme is robust against room noise and video tape recording.

Bender *et al.* [6] propose several techniques for watermarking which are applicable to audio. They call the techniques spread-spectrum coding, echo coding, and phase coding. Direct sequence spread-spectrum coding is performing biphase shift keying on a carrier wave by using an encoded binary string and pseudorandom noise. The code introduces perceptible noise into the original sound signal, but by using adaptive coding and redundant coding the perceptible noise can be reduced. Echo coding is a method which employs multiple decaying echos to place a peak in the cepstrum at a known location. The result is that moderate amounts of data can be hidden in a form that is fairly robust versus "analog" transmission. Phase coding is a method that employs the phase information as a data space. For the encoding, a Fourier transform is applied and the phase values of each frequency component are lined up as a matrix; binary information can be embedded into this matrix by modifying the phase component. Since the human HAS is not very sensitive to the distortion to the phase of the sound, it can be used to encode data without introducing much audible distortion to the original sound.

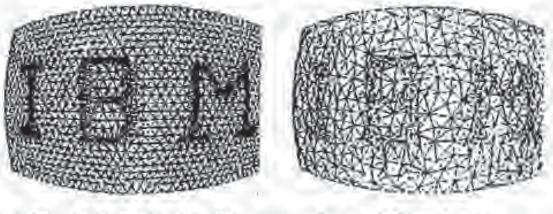


Fig. 11. Embedding of visible watermarks into 3-D meshes by local variation of the mesh density. (Figure taken with kind permission from [94].)

## VIII. WATERMARKING OF OTHER MULTIMEDIA DATA

Most watermarking research, publications, and products are dedicated to images. Less has been published on video, audio, and formatted text watermarking, and even less on watermarking of other media. However, the underlying basic ideas are certainly applicable to almost all kinds of digital data.

Ohbuchi *et al.* [94], [95] have proposed methods for embedding visible and invisible watermarks into 3-D polygonal models. Such models comprise primitives like points, lines, polygons, and polyhedrons, which are attributed by their geometry and their topology. Ohbuchi *et al.* propose to modify geometry or topology for watermarking. In detail, they propose two different methods for embedding of invisible watermarks for models consisting of triangular meshes. The first method pseudorandomly selects sets of four adjacent triangles and embeds information by displacing the vertices of the four triangles in a specific way by up to 1% of the shortest edge of the rectangular bounding box of the entire 3-D model. The authors claim that the modifications are imperceptible and that the method is resistant to cropping if the watermark information is repeated several times over the 3-D model and to local deformation. The second method pseudorandomly selects tetrahedron from the mesh and embeds information in the volume ratio of consecutive tetrahedron by modification of vertices. This method is robust against cropping and local deformation. A third method embeds visible watermarks into meshes by local variation of the mesh density, as shown in Fig. 11.

The emerging video compression standard MPEG-4 features additional functionalities, besides common video compression, such as model-based animation of 3-D head models using so-called facial animation parameters (FAP's). These are parameters like "rotate head," "open mouth," or "raise right corner-lip." The head model used at the receiver is either a predefined generic head and face model or a particular model that can be transmitted using so-called facial definition parameters (FDP's). The tool for face animation allows the compression of head-and-shoulder scenes, for example, in video telephony applications, with bit rates below 1000 bits/s. In [46], Hartung *et al.* propose a spread-spectrum method for watermarking of MPEG-4 FAP's. The watermarks are additively embedded into the animation parameters. Smoothing of the spread-spectrum watermark by low-pass filtering and an adaptive amplitude attenuation prevents visible distortions of the animated head models. The watermarks can be retrieved by correlation from the watermarked parameters, but also from video sequences showing 3-D head models animated with the watermarked parameters, even after modifications such as block-based compression. Fig. 12 shows examples of video frames from a sequence rendered from a 3-D head model and animation parameters. In this case, the parameters first have to be estimated from the sequence. An interesting point is that the watermark is not contained in the waveform representation of the depicted object (the pixels), but in the semantics (the way the head and face move).

## IX. WATERMARK APPLICATIONS, SECURITY, ROBUSTNESS, AND CRYPTOANALYSIS

### A. Applications

We have already seen in Section III that the requirements and the design constraints for watermarking technologies strongly depend on the final application. For obvious reasons there is no "universal" watermarking method. Although watermarking methods have to be robust in general, different levels of required robustness can be identified depending on the specific application-driven requirements.

In authentication applications, the watermarks have to resist only to certain attacks. Among all possible watermarking applications, authentication watermarks require the lowest level of robustness. The purpose of such watermarks is to authenticate the data content. For example, data can be watermarked such that the watermark can accommodate lossy compression, but they are destroyed as soon as the data are manipulated in a different way.

Applications such as data monitoring and tracking require a higher level of robustness. The main purpose is to detect or identify stored or transmitted data. Examples are automatic monitoring of radio broadcast for billing purposes or identification of images on the World Wide Web with the help of web crawlers. For such applications, the watermarks have to be easily extractable and must be reasonably robust, for example, against standard data processing like format conversion and compression.

In fingerprinting applications, watermarks are embedded that identify the recipient of each individual distributed copy. The purpose is to have a means to trace back pirated copies to the recipient who pirated it. Fingerprinting applications require a very high level of robustness against data processing and malicious attacks.

Watermarking for copyright protection is used to resolve rightful ownership and requires the highest level of robustness. However, robustness alone is not sufficient for such applications. For example, if different watermarks are embedded in the same data, it must still be possible to identify the first, authoritative, watermark. Hence, additional design requirements besides mere robustness apply, as discussed below.

In the following, we go into more details on how to resist malicious attacks and elaborate on design constraints for copyright protection applications of watermarking.


(a) (b)

Fig. 12. Example frame from a video sequence rendered from (a) a 3-D head model and watermarked animation parameters and (b) a similar frame after subsequent MPEG-2 video compression at 600 kbit/s.

### *B. Watermark Robustness*

Robustness against attacks is a major watermarking requirement. Absolute robustness against all possible attacks and their combinations may be impossible to achieve. Thus, the practical requirement is that a successful attack must impair the host data to the point of significantly reducing its commercial value before the watermark is impaired so much that it cannot be recovered. In fact, with appropriate design, fairly high robustness can be achieved, but it should be pointed out that robustness always has to be traded against watermark data rate and imperceptibility, and the optimum tradeoff depends on the application.

*1) Classification of Attacks:* Following the classification in [50], four different types of attacks can be identified.

1) "Simple attacks" (other possible names include "waveform attacks" and "noise attacks") are conceptually simple attacks that attempt to impair the embedded watermark by manipulations of the whole watermarked data (host data plus watermark) without an attempt to identify and isolate the watermark. Examples include linear and general nonlinear filtering, waveform-based compression (JPEG, MPEG), addition of noise, addition of an offset, cropping, quantization in the pixel domain, conversion to analog, and gamma correction.
2) "Detection-disabling attacks" (other possible names include "synchronization attacks") are attacks that attempt to break the correlation and to make the recovery of the watermark impossible or infeasible for a watermark detector, mostly by geometric distortion like zooming, shift in spatial or temporal (for video) direction, rotation, shear, cropping, pixel permutations, subsampling, removal or insertion of pixels or pixel clusters, or any other geometric transformation of the data.
3) "Ambiguity attacks" (other possible names include "deadlock attacks," "inversion attacks," "fake-watermark attacks," and "fake-original attacks") are attacks that attempt to confuse by producing fake original data or fake watermarked data [54]. An example is an inversion attack [30]–[32] that attempts to discredit the authority of the watermark by embedding one or several additional watermarks such that it is unclear which was the first, authoritative watermark.
4) "Removal attacks" are attacks that attempt to analyze the watermarked data, estimate the watermark or the host data, separate the watermarked data into host data and watermark, and discard only the watermark. Examples are collusion attacks [121], denoising, certain nonlinear filter operations [81], or compression attacks using synthetic modeling of the image (e.g., using texture models or 3-D models). Also included in this group are attacks that are tailored to a specific watermarking scheme and combat it by exploiting conceptual cryptographic weaknesses of the scheme that make it vulnerable to a specific attack.

It should be noted that the transitions between the groups are sometimes fuzzy and that some attacks do not clearly belong to one group. Collusion attacks could be argued to be a group of its own, since they require, unlike the other attacks, more than one differently watermarked copy of the data. However, since they attempt to reconstruct the unwatermarked original host data, and thus remove the watermark(s), the classification as a "removal attack" holds.

In the following, remedies are given that make watermarks more robust against malicous attacks.

*2) Remedies Against Simple, Waveform-Based Attacks:* As already mentioned, noise-like distortions, for example, due to lossy compression, result in a distorted watermark signal in the watermark recovery or verification process. There are two main remedies against such attacks: increasing the embedding strength or applying redundant embedding. Increasing the embedding strength is straightforward and efficient in many cases, especially if appropriate masking according to the properties of human perception is used to determine the maximum allowable embedding strength. Redundant embedding can be performed in many ways. In the spatial domain it might consist of embedding a watermark many times and then taking a majority vote in the recovery process. A more efficient technique could include the use of error-correcting codes [52], possibly even with soft-decision decoding [51]. Both increasing the watermark strength and introducing redundancy either increase the watermark visibility/audibility or decrease the watermark data rate. Further, as pointed out before, it should be noted that there is a tradeoff between watermark robustness on one hand and watermark imperceptibility and watermark data rate on the other hand.

*3) Geometrical Distortions and Remedies:* Watermarks are typically most vulnerable to geometrical distortions. The reason is that, for most proposed watermarking methods, the watermark detector has to know the exact position of the embedded watermark. Geometrical distortions tend to destroy the synchronization such that watermark embedding and watermark detection are misaligned and do not fit anymore.

Simple geometric attacks include affine transforms, clipping, and cropping. Remedies against such attacks are difficult if the watermarking algorithm has not explicitly been designed to withstand such attacks [114]. For this "simple" geometrical attacks, the challenge consists of finding the original watermark reference within the host data. For watermarking schemes which require the original image to recover the watermark this may not be a real problem, since the geometrical distortion can be estimated from the two images and inverted. If the watermarking scheme does not have the original data available for the watermark recovery, many schemes still allow the reference recovery by using a full search over all possible manipulations using some kind of correlation criteria between the image and the watermark modulation sequence. If the geometrical distortion consists of simple cropping, translation, or rotation, this process is feasible. However, if the attack consists of any affine transform this becomes very intensive computationally. Another way to resist geometrical attacks is based on embedding a watermark reference within the host data. Gruhl and Bender [45] propose embedding invisible crosses into the image by modifying the LSB image plane. Later detection of the crosses allows exact determination of the undergone attack and thus its reversal. If resistance to cropping has also to be assured, the row and column information can be encoded in addition to the crosses. One simple way of doing so would, for example, consist of changing the horizontal and vertical spacing between crosses depending on the location within the image. Although fully functioning, this system is not very robust since the reference can very easily be removed or destroyed. Another example is the embedding of sinusoidal patterns in the color channel using a visibility metric to ensure invisibility, as proposed by Fleet and Heeger [42]. An extension of the method of Gruhl and Bender has been proposed by Kutter [76] in which a spatial watermark pattern is embedded four times into the host image by using predetermined horizontal and vertical shifts. In the recovery process an autocorrelation function of an estimated watermark pattern can be computed to determine the affine distortion. Applying the inverse transform then allows full recovery of the watermark. A more sophisticated geometrical attack is based on jittering [70], [100], [138]. Jittering cuts the data set in small chunks, than randomly removes or duplicates small pieces and then sticks the small chunks back together. If done in a smart way, this alteration introduces only little or even no perceptible artifacts. This attack has proven to be very efficient in removing watermarks for many algorithms. Remedies exist against this attack, depending on the algorithm. For example, the method proposed by Kutter *et al.* [74] resists jittering if the image under inspection is low-pass filtered before the watermark extraction process. For other methods this remedy might work as well.

*4) Watermark Removal Attacks and Remedies:* Collusion attacks are attacks that use several copies of the same host data with different embedded watermarks. Several types of collusion attacks have been examined by Cox *et al.* [27] and Stone [121]. In the following, a watermark observation refers to a watermarked data representation in any domain, e.g., spatial or frequency domain. The first attack is called statistical averaging, in which a new data set is created by taking the average of all available watermark observations. A second attack creates a new data set by taking the average of the minimum and maximum of all watermark observations. The third approach is based on introducing negative correlation as follows:

$$\hat{w} = \begin{cases} w_{\max}, & \text{if } w_{\text{median}} \leq \dfrac{(w_{\max} - w_{\min})}{2} \\ w_{\min}, & \text{otherwise} \end{cases} \tag{35}$$

where $w_{\text{median}}$, $w_{\min}$, and $w_{\max}$ are the median, minimum, and maximum of the all watermark observations. Stone shows that for the image watermarking scheme proposed by Cox *et al.* [27] and a watermark with uniform distribution, at least four watermark observations are required for a successful attack. In general, all these statistical attacks can successfully destroy embedded watermarks even if only a few watermarked data sets are available. Another collusion attack interleaves the different watermarked copies of the same data [121]. Small parts of different watermarked data sets are taken and reassembled in a new data set. A remedy against collusion attacks is to limit the available number of watermarked copies. Alternatively, it has been proposed to use collusion-secure codes to design watermarks [9], [10]. The drawback is that the code lengths increase exponentially with the number of codes.

If the watermark detector device is available, the Oracle attack, first proposed by Perrig [98] and further developed by Cox and Linnartz [28], [29], can be used to destroy the embedded watermark. Such a scenario is, for example, possible in copy control systems for digital media, such as the DVD. The watermark detector can be used to experimentally deduce its behavior and then destroy the watermark. Although commonly believed that this approach involves an extremely high complexity, the authors illustrate that this is not true and claim the complexity to be of order $O(N)$, where $N$ is the number of data samples, for most watermarking system. If the watermark inserter is available, another attack is based on predistorting the original data set. The difference between the watermarked data set and original data set is used to predistort the original data set through subtraction. The newly watermarked predistorted data set is then very unlikely to contain the watermark. One remedy against a predistortion attack is based on encryption using a random session key. Given a binary watermark $W$ to be embedded into a set of data, it is first encrypted using a random encryption key $k$ resulting in $W_k$. The key is then appended to the encrypted watermark to give the new watermark $W_{K'}$, which is then embedded into the host data set. The watermark detector can recover the embedded watermark and decrypt it. The predistortion attack fails because the watermark inserter is not deterministic anymore due to the fact that the embedded watermark changes each time.

A histogram-based attack called *Twin Peaks* for fixed depth bimodal watermarks has been proposed by Maes [88]. To illustrate the concept of the attack, let us consider an image histogram with a peak at the intensity level $P$. Further, let us assume that the image was watermarked with a uniformly distributed watermark with a bimodal amplitude of $\pm d$. In this case, the watermarking process maps 50% of the values from $P$ to $P + d$, and the other 50% from $P$ to $P - d$. The peak in the original histogram at intensity $P$ is therefore replaced by two peaks at intensities $P - d$ and $P + d$ (hence the name *Twin Peaks*), both having half the height of the original peak. By looking at the histogram of a watermarked image, it is possible to determine the embedded watermark by detecting close by peaks with similar amplitude. The original value may then be estimated and substituted into the watermarked image in order to destroy the embedded watermark. Based on this idea, the author show how to successfully destroy embedded watermarks. The performance of the attack may be improved when a prediction of the embedded watermark is used instead of the watermarked image. The prediction is computed by filtering the image with a high-pass filter which can be seen as taking the difference between a pixel value and the local mean computed in a squared wind of size 3 × 3.

### C. Remedies Against Watermark Ambiguities

As mentioned at the beginning of this section, to resolve rightful ownership, it must be possible to determine the authoritative watermark in case several watermarks are present in a data set.

*1) Timestamps:* To determine who first signed a set of data, timestamps (provided by trusted third parties) should be used [117], [149]. Let $X$ be the data to be time stamped and $H$ the corresponding hash value. The owner sends an official request $R_n = (H_n, I_n)$, where $I_n$ is the owners identification string, to an official third party time stamping service (TSS). The TSS produces a timestamp $TS_n$

$$TS_n = S_k(n, I_n, H_n, T_n, I_{n-1}, H_{n-1}, T_{n-1}, L_n) \quad (36)$$

where $n$ is the request number, $T_n$ the time of the request, and $S_k$ indicates that the message is signed with the public key of TSS. $L_n$ is known as the linking string defined as

$$L_n = H(I_{n-1}, H_{n-1}, T_{n-1}, L_{n-1}) \quad (37)$$

and is used to avoid that the timestamp requester and the TSS collude to produce any timestamp they want. The TSS then waits for the next request and returns the new identification $I_{n+1}$ of the originator. If someone challenges a timestamp $TS_n$, the owner can prove that is was stamped after and before those by $I_{n-1}$ and $I_{n+1}$, respectively. If their documents are also called in question they can get in touch with $I_{n-2}$ and $I_{n+2}$, and so on.

Because digital time stamping involves a trusted third party, the question might arise why to use watermarking in combination with timestamping since this is very similar to traditional copyright registration and protection of copyright laws.

*2) Noninvertible Watermarks:* Until the publications of Craver *et al.* [30]–[32] it was believed that with the help of the original, nonwatermarked data set one can easily prove rightful ownership. Craver *et al.* showed that having the original is not sufficient and introduced the expression of invertible watermarking schemes. Given an original data set $I_o$ to be watermarked with $W_1$

$$\hat{I}_o = I_o \oplus W_1 \quad (38)$$

where $\hat{I}_o$ is the watermarked original and the operator $\oplus$ represents watermark insertion. Craver *et al.* showed that certain watermarking methods are invertible and allow reverse engineering to produce a counterfeit original

$$I_c = \hat{I}_o \ominus W_2 \quad (39)$$

where $I_c$ it the counterfeit original and $\ominus$ the inversion process. Let further assume that $D$ is a watermark decoder function with a binary output of "0" and "1" for watermark absent and watermark present, respectively. This scenario creates an ownership deadlock because the rightful owner can show that his watermark is presents in the signed data and counterfeit original

$$\begin{aligned} D(I_o, \hat{I}_o, W_1) &= 1 \\ D(I_o, I_c, W_1) &= 1. \end{aligned} \quad (40)$$

However, the attacker can also show that his watermark $W_2$ is present in the watermarked original, as well as in the original

$$\begin{aligned} D\left(I_c, \hat{I}_o, W_2\right) &= 1 \\ D(I_o, I_c, W_2) &= 1. \end{aligned} \quad (41)$$

Hence it is not possible to resolve rightful ownership since all claims from both parties are legally speaking equivalent. Some watermarking techniques are inherently invertible and the question is how to make them noninvertible or how to avoid this problem. Meanwhile, several methods have been devised to construct noninvertible watermarks [92], [110], [128]. The general idea in most methods is to make watermarks noninvertible by making them signal dependent, for example, by using one-way hash functions. In this case, it is computationally infeasible for an attacker to create a counterfeit original because it depends on

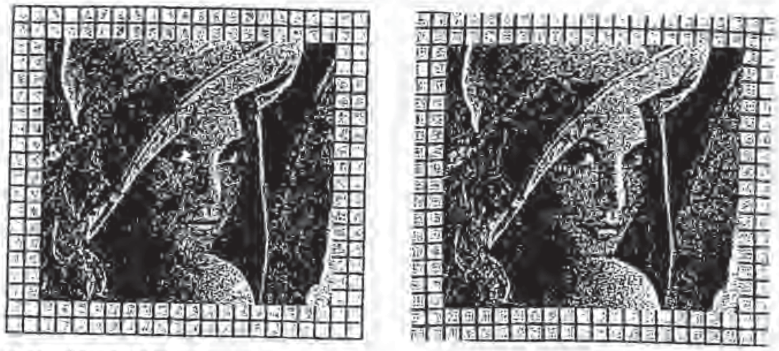


Fig. 13. Demonstration of the StirMark 2.2 attack.

the watermark, which in turn depends on the counterfeit original which is not yet existing.

It should also be noted that in applications where the owner of the data is undisputed, like, for example, in labeling applications where a serial number is embedded into different copies of distributed data, the above concerns do not apply.

### *D. Robustness Test Utilities and Watermark-Removal Software*

Similar to conditional access and copy-prevention mechanisms, the existence of watermarking technology and its potential possibilities have stimulated individuals to come up with attempts to defeat watermarking. Examples are publicly available tools to test the robustness of image watermarking techniques. Unzign [138] is a utility that works for images in JPEG format. In version 1.1, Unzign introduces pixel jittering in combination with a slight image translation. For many proposed watermarking techniques, the embedded watermarks are efficiently destroyed. However, besides removing the watermark, Unzign version 1.1 introduces severe artifacts. An improved version 1.2 has been released. Although the artifacts were decreased, its watermark destruction capability decreased as well.

StirMark [70], [100] is a simple generic tool to test the robustness of image watermarking techniques. It simulates resampling to emulate a printing–scanning procedure and applies minor geometric distortions (stretching, shearing, shifting, and rotation) followed by resampling and bilinear or Nyquist interpolation. In addition, small and smoothly distributed errors are introduced into all sample values. Applying StirMark only once introduces a practically unnoticeable quality loss in the image. The author claims that his tool removes all current watermarks. Fig. 13 demonstrates the affect of the StirMark attack on a test image containing a grid and a natural image, and its StirMark 2.2 attacked version. From visual inspection, it can be confirmed that the effect of the attack is not visually annoying in the image, and is only evident in the grid. However, this attack is quite successful if the watermarking method does not account for it [50].

## X. THE FUTURE OF DIGITAL WATERMARKING

The interest in watermarking technology is high, both from academia and industry. The interest from academia is reflected in the number of publications on watermarking and in the fact that conferences on watermarking and data hiding are being held. The interest from industry is evident in the number of companies in the field that have been founded within the past few years.

Besides research activities in universities and industry, several international research projects funded by the European Community have the goal to develop practical watermarking techniques. TALISMAN [61] (ACTS project AC019, "Tracing Authors' rights by labeling image services and monitoring access network") aims to provide European Union service providers with a standard copyright mechanism to protect digital products against large scale commercial piracy and illegal copying. The expected output of TALISMAN is a system for protecting video sequences through labeling and watermarking. OCTALIS [60] (ACTS project P119, "Offer of Content through Trusted Access Links") is the follow-up project of TALISMAN and OKAPI with the main goal of integrating a global approach to equitable conditional access and efficient copyright protection and to demonstrate its validity on large scale trials on the Internet and European Broadcasting Union (EBU) network.

International standardization consortia are also interested in watermarking techniques. The emerging video compression standard MPEG-4 (ISO/IEC 14496), for example, provides a framework that allows the easy integration with encryption and watermarking. The DVD industry standard will contain copy control and copy protection mechanisms that use watermarking to signal the copy status of multimedia data, like "copy once" or "do not copy" flags.

Despite the many efforts that are underway to develop and establish watermarking technology, watermarking is still not a fully mature and understood technology, and a lot of questions are not answered yet. Also, the theoretical fundamentals are still weak, and most systems are designed heuristically.

Another drawback is that fair comparisons between watermarking systems are difficult [75]. As long as methods and system implementations are not evaluated in a consistent manner using sophisticated benchmarking methods, the danger exists that weak and vulnerable systems and *de facto* standards are produced that result in spectacular failures and discredit the entire concept.

Thus, the expectations into watermarking should be realistic. It should always be kept in mind that every watermarking system involves a tradeoff between robustness, watermark data rate (payload), and imperceptibility. The invisible 10 000-bit-per-image watermark that resists all attacks whatsoever is an illusion (realistic numbers are approximately two orders of magnitude lower). Even when designed under realistic expectations, watermarks offer robustness against nonexperts but may still be vulnerable to attacks by experts.

Although proof of ownership was the initial thrust for the technology, it seems that there is a long way to go before watermarking will be accepted as a proof in court, and it is likely enough that this may never happen. In copyright-related applications, watermarking must be combined with other mechanisms like encryption to offer reliable protection.

Still, there exist enough applications where watermarking can provide working and successful solutions. Specifically for audio and video it seems that watermarking technology will become widely deployed. The DVD industry standard, as an example, will use watermarking for the copy protection system. Similarly, there exist plans to use watermarking for copy protection for Internet audio distribution. Broadcast monitoring using watermarking is another application that will probably widely be deployed for both audio and video.

Whether the development of watermarking technology will become a success story or not is an interesting yet unclear question. Watermarking technology will evolve, but attacks on watermarks as well. Careful overall system design under realistic expectations is crucial for successful applications.

## XI. CONCLUSIONS

In this overview paper, we reviewed the most important aspects, design requirements, system issues, and techniques for digital watermarking. The historical roots of digital watermarking derive mainly from steganography, the art of data hiding. Although digital watermarking and steganography are in some sense similar, the main difference lies in the notion of robustness for digital watermarks. Watermark robustness is one of the major design issues, besides imperceptibility. We have shown that the various digital watermarking applications, such as data tracking, data monitoring, and copyright protection, result in corresponding design issues and algorithm requirements. Some schemes require the original data set in order to recover an embedded watermark and others do not. Further, in some publications methods are proposed that allow full watermark extraction, whereas in other publications techniques are presented which only allow verification if a given watermark is present in the data under investigation. We have emphasized that these two approaches are inherently equivalent in that a watermark-extraction scheme can be transformed into a watermark-verification scheme and vice versa. Although often associated to still images, video, and audio, digital watermarking is also applicable to other digital data such as text, 3-D meshes, or face animation parameters. We have elaborated on numerous watermarking techniques for still images, video, audio, text, and other multimedia data. It has been pointed out that a majority of techniques are inherently similar and based on modulation with a PN signal, often in combination with masking, for the embedding process and some kind of hypothesis testing using correlation in the watermark recovery process. Designing watermarking methods does not only have to consider robustness against standard data processing, but also robustness against malicious attacks. Several classes of attacks have been outlined, and remedies were given to make watermarks attack resistant. As a general statement, it can be said that watermarks should be sufficiently overdesigned and contain enough redundancy to ensure resilience against attacks. For copyright enforcement, additional aspects have to be considered. One problem is to prove who first watermarked data if several watermarks are present in the data. Solutions to this problem might consist of digital time stamping or watermark registration. Further, it has been shown that robustness is not sufficient to resolve rightful ownership, even if the original data are available. In addition, the used watermarking method needs to be noninvertible. Several techniques have been proposed to render invertible methods noninvertible, including hashing and time stamping. Although working systems are already available, research in digital watermarking has to continue. There is a huge demand from content providers and IPR owners. The market is currently far from being saturated and many more companies are expected to be founded in the near future. The question whether digital watermarks will be used as legal proof in court is not yet decided and difficult to answer. There are, however, other applications, like multimedia copy protection systems and data broadcast monitoring, where we will see watermarking in operation.

D

## ACKNOWLEDGMENT

The authors would like to thank Dr. I. Cox, Prof. E. Delp, Dr. A. Herrigel, Dr. T. Kalker, Prof. M. Kobayashi, D. Kundur, S. Moskowitz, Prof. I. Pitas, Prof. T. Pun, and Dr. J. Zhao for sharing their views on the future of watermarking technology. Significant parts of Section X are a summary of their contributions. The authors would further like to thank Dr. J. K. Su and the anonymous reviewers for their suggestions which helped to improve the quality of the paper. The second author thanks Prof. Ebrahimi, Swiss Federal Institute of Technology, Lausanne, for introducing him to the presented topic and is grateful for the technical discussions, insights, and hints.

## REFERENCES

[1] R. J. Anderson and F. Petitcolas, "On the limits of steganography," *IEEE J. Select. Areas Commun. (Special Issue on Copyright and Privacy Protection)*, vol. 16, pp. 474–481, May 1998.

[2] M. Barni, F. Bartolini, V. Cappellini, and A. Piva, "A DCT-domain system for robust image watermarking," *Signal Processing (Special Issue on Watermarking)*, vol. 66, no. 3, pp. 357–372, May 1998.

[3] M. Barni, F. Bartolini, V. Cappellini, A. Piva, and F. Rigacci, "A M.A.P. identification criterion for DCT-based watermarking," in *Proc. Europ. Signal Processing Conf. (EUSIPCO '98)*, Rhodes, Greece, Sept. 1998.

[4] P. Bas and J.-M. Chassery, "Using fractal code to watermark images," in *Proc. Int. Conf. Image Processing (ICIP)*, vol. 1, Chicago, IL, 1998.

[5] P. Bassia and I. Pitas, "Robust audio watermarking in the time domain," in *Proc. European Signal Processing Conf. (EUSIPCO 98)*, Rhodes, Greece, Sept. 1998.

[6] W. Bender, D. Gruhl, and N. Morimoto, "Techniques for data hiding," in *Proc. SPIE*, vol. 2420, San Jose, CA, Feb. 1995, p. 40.

[7] D. Benham, N. Memon, B.-L. Yeo, and M. Yeung, "Fast watermarking of DCT-based compressed images," in *Proc. Int. Conf. Image Science, Systems, and Technology (CISST '97)*, Las Vegas, NV, June 1997, pp. 243–253.

[8] F. M. Boland, J. J. K. Ó Ruanaidh, and W. J. Dowling, "Watermarking digital images for copyright protection," in *Proc. Int. Conf. Image Processing and Its Applications*, vol. 410, Edinburgh, U.K., July 1995.

[9] D. Boneh and J. Shaw, "Collusion-secure fingerprinting for digital data," in *Advances in Cryptology—Proc. CRYPTO '95* (Lecture Notes in Computer Science), vol. 963, Don Coppersmith, Ed. Berlin, Germany: Springer, 1995, pp. 452–465.

[10] ——, "Collusion-secure fingerprinting for digital data," *IEEE Trans. Inform. Theory*, vol. 44, pp. 1897–1905, Sept. 1998.

[11] L. Boney, A. H. Tewfik, and K. H. Hamdy, "Digital watermarks for audio signals," in *Proc. EUSIPCO 1996*, Trieste, Italy, Sept. 1996.

[12] A. Bors and I. Pitas, "Embedding parametric digital signatures in images," in *EUSIPCO-96*, Trieste, Italy, Sept. 1996.

[13] ——, "Image watermarking using DCT domain constraints," in *Proc. Int. Conf. Image Processing (ICIP)*, Lausanne, Switzerland, Sept. 1996.

[14] J. Brassil, S. Low, N. Maxemchuk, and L. O'Gorman, "Electronic marking and identification techniques to discourage document copying," *IEEE J. Select. Areas Commun.*, vol. 13, pp. 1495–1504, Oct. 1995.

[15] ——, "Hiding information in document images," in *Proc. 29th Annu. Conf. Information Sciences and Systems (CISS 95)*, Johns Hopkins Univ., Baltimore, MD, Mar. 1995, pp. 482–489.

[16] G. W. Braudaway, K. A. Magerlein, and F. C. Mintzer, "Color correct digital watermarking of images," U.S. Patent 5 530 759, June 1996.

[17] O. Bruyndonckx, J. J. Quisquater, and B. Macq, "Spatial method for copyright labeling of digital images," in *Proc. IEEE Workshop Nonlinear Signal and Image Processing*, Halkidiki, Greece, June 1995.

[18] S. Burgett, E. Koch, and J. Zhao, "A novel method for copyright labeling digitized image data," Fraunhofer Inst. Comput. Graphics, Darmstadt, Germany, Tech. Rep., Sept. 1994.

[19] C. Busch, W. Funk, and S. Wolthusen, "Digital watermarking: From concepts to real-time video applications," *IEEE Comput. Graphics Applicat.*, pp. 25–35, Jan. 1999.

[20] G. Caronni, "Ermitteln unauthorisierter Verteiler von maschinenlesbaren Daten," ETH, Zürich, Switzerland, Tech. Rep., Aug. 1993.

[21] ——, "Assuring ownership rights for digital images," in *Proc. VIS 95, Session "Reliable IT Systems,"* Vieweg, Germany, 1995.

[22] B. Chen and G. W. Wornell, "Digital watermarking and information embedding using dither modulation," in *Proc. IEEE Workshop Multimedia Signal Processing*, Los Angeles, CA, Dec. 1998.

[23] ——, "Dither modulation: A new approach to digital watermarking and information embedding," in *IS&T/SPIE's 11th Annu. Symp., Electronic Imaging '99: Security and Watermarking of Multimedia Contents*, vol. 3657, San Jose, CA, Jan. 1999.

[24] ——, "An information-theoretic approach to the design of robust digital watermarking systems," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing 1999 (ICASSP '99)*, Phoenix, AZ, Mar. 1999.

[25] G. Cooper and C. McGillem, *Modern Communications and Spread Spectrum*. New York: McGraw-Hill, 1986.

[26] I. Cox, J. Kilian, T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for images, audio and video," in *Proc. IEEE Int. Conf. Image Processing (ICIP 96)*, Lausanne, Switzerland, Sept. 1996.

[27] ——, "Secure spread spectrum watermarking for images, audio, and video," NEC Res. Inst., Princeton, NJ, Tech. Rep. 95-10, 1995.

[28] I. J. Cox and J.-P. Linnartz, "Some general methods for tampering with watermarks," *IEEE J. Select. Areas Commun. (Special Issue on Copyright and Privacy Protection)*, vol. 16, pp. 587–593, May 1998.

[29] I. J. Cox, J.-P. Linnartz, and T. Shamoon, "Public watermarks and resistance to tampering," in *Proc. Int. Conf. Image Processing (ICIP)*, 1997.

[30] S. Craver, N. Memon, B.-L. Yeo, and M. Yeung, "Can invisible watermarks resolve rightful ownerships?," IBM, IBM Res. Rep. RC 20509, July 1996.

[31] ——, "On the invertibility of invisible watermarking techniques," in *Proc. IEEE Int. Conf. Image Processing 1997 (ICIP '97)*, vol. 1, Santa Barbara, CA, Oct. 1997, pp. 540–543.

[32] ——, "Resolving rightful ownerships with invisible watermarking techniques: Limitations, attacks, and implications," *IEEE J. Select. Areas Commun. (Special Issue on Copyright and Privacy Protection)*, vol. 16, pp. 573–586, May 1998.

[33] V. Darmstaedter, J.-F. Delaigle, D. Nicholson, and B. Macq, "A block based watermarking technique for MPEG-2 signals: Optimization and validation on real digital TV distribution links," in *Proc. European Conf. Multimedia Applications, Services, and Techniques—ECMAST '98*, Berlin, Germany, May 1998.

[34] P. Davern and M. Scott, "Fractal based image steganography," in *Lecture Notes in Computer Science: Information Hiding*, vol. 1174. Berlin, Germany: Springer, 1996, pp. 279–294.

[35] F. Deguillaume. (1999, Jan.). Video watermarking—MPEG-2 video samples used for 3D-DFT video watermarking tests. [Online]. Available WWW: http://cuiwww.unige.ch/deguillo/WM/wm.html.

[36] F. Deguillaume, G. Csurka, J. Ó Ruanaidh, and T. Pun, "Robust 3D DFT video watermarking," in *IS&T/SPIE's 11th Annu. Symp., Electronic Imaging '99: Security and Watermarking of Multimedia Contents*, vol. 3657, San Jose, CA, Jan. 1999.

[37] J. F. Delaigle, D. De Vleeschouwer, and B. Macq, "Low cost perceptive digital picture watermarking method," in *Proc. ECMAST'97*, Milan, Italy, May 1997, pp. 153–167.

[38] G. Depovere, T. Kalker, and J.-P. Linnartz, "Improved watermark detection reliability using filtering before correlation," in *Proc. IEEE Int. Conf. Image Processing 1998 (ICIP 98)*, Chicago, IL, Oct. 1998.

[39] J. Dittmann, M. Stabenau, and R. Steinmetz, "Robust MPEG video watermarking technologies," in *Proc. ACM Multimedia '98*, Bristol, U.K., Sept. 1998.

[40] R. C. Dixon, *Spread Spectrum Systems with Commercial Applications*. New York: Wiley, 1994.

[41] O. Emery, "Des filigranes du papier," *A.T.I.P. Bull., Bull. de l'Association Technique de l'Industrie Papetiere*, vol. 12, no. 6, pp. 185–188, 1958.

[42] D. Fleet and D. Heeger, "Embedding invisible information in color images," in *Proc. IEEE Int. Conf. Image Processing 1997 (ICIP '97)*, Santa Barbara, CA, vol. 1, Oct. 1997, pp. 532–535.

[43] P. G. Flikkema, "Spread-spectrum techniques for wireless communications," *IEEE Signal Processing Mag.*, vol. 14, pp. 26–36, May 1997.

[44] J. Fridrich, "Methods for data hiding," State Univ. New York, Binghamton, Tech. Rep., 1997.

[45] D. Gruhl and W. Bender. (1995). Affine invariance. [Online]. Available WWW: http://nif.www.media.mit.edu/DataHiding/affine/affine.html.

[46] F. Hartung, P. Eisert, and B. Girod, "Digital watermarking of MPEG-4 facial animation parameters," *Comput. Graphics*, vol. 22, no. 3, pp. 425–435, 1998.

[47] F. Hartung and B. Girod, "Digital watermarking of raw and compressed video," in *Proc. SPIE Digital Compression Technologies and Systems for Video Commun.*, vol. 2952, Oct. 1996, pp. 205–213.

[48] ——, "Fast public-key watermarking of compressed video," in *Proc. IEEE Int. Conf. on Image Processing 1997 (ICIP '97)*, vol. 1, Santa Barbara, CA, Oct. 1997, pp. 528–531.

[49] ——, "Digital watermarking of uncompressed and compressed video," *Signal Processing (Special Issue on Copyright Protection and Access Control for Multimedia Services)*, vol. 66, no. 3, pp. 283–301, 1998.

[50] F. Hartung, J. K. Su, and B. Girod, "Spread spectrum watermarking: Malicious attacks and counterattacks," in *Proc. SPIE Security and Watermarking of Multimedia Contents 99*, San Jose, CA, Jan. 1999.

[51] F. Hartung, "Digital watermarking and fingerprinting of uncompressed and compressed video," Ph.D. dissertation, Telecommun. Lab., Univ. Erlangen–Nuremberg, Erlangen, Germany, 1999.

[52] J. R. Hernández, F. Pérez-González, and J. M. Rodríguez, "The impact of channel coding on the performance of spatial watermarking for copyright protection," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing 1998 (ICASSP 98)*, Seattle, WA, vol. 5, May 1998, pp. 2973–2976.

[53] J. R. Hernández, F. Pérez-González, J. M. Rodríguez, and G. Nieto, "Performance analysis of a 2-d multipulse amplitude modulation scheme for data hiding and watermarking still images," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 510–524, 1998.

[54] M. Holliman and N. Memon, "Counterfeiting attacks on linear watermarking schemes," in *Proc. IEEE Multimedia Systems '98, Workshop Security Issues in Multimedia Systems*, Austin, TX, June 1998.

[55] C.-T. Hsu and J.-L. Wu, "Hidden signatures in images," in *Proc. IEEE Int. Conf. Image Processing (ICIP 96)*, Lausanne, Switzerland, Sept. 1996, pp. 223–226.

[56] ——, "Digital watermarking for video," in *Proc. of DSP'97*, Santorini, Greece, July 1997.

[57] C.-T. Hsu, "Digital watermarking for images and videos," Ph.D. dissertation, Commun. Multimedia Lab., National Taiwan Univ., 1997.

[58] H. Inoue, A. Miyazaki, A. Yamamoto, and T. Katsura, "A digital watermark based on the wavelet transform and its robustness on image compression," in *Proc. Int. Conf. Image Processing (ICIP)*, Chicago, IL, 1998.

[59] A. Johnson and M. Biggar, "Digital watermarking of video/image content for copyright protection and monitoring," ISO, ISO Doc. ISO/IEC JTC1/SC29/WG11 MPEG97/M2228, July 1997.

[60] P. Jones. Octalis. [Online]. Available WWW: http://www.cordis.lu/esprit/src/octalis.htm.

[61] ——. Talisman. [Online]. Available WWW: http://www.cordis.lu/esprit/src/talisman.htm.

[62] F. Jordan, M. Kutter, and T. Ebrahimi, "Proposal of a watermarking technique for hiding/retrieving data in compressed and decompressed video," ISO/IEC Doc. JTC1/SC29/WG11 MPEG97/M2281, July 1997.

[63] T. Kalker, private communication.

[64] ——, "Watermark estimation through detector observations," in *Proc. IEEE Benelux Signal Processing Symposium '98*, Leuven, Belgium, Mar. 1998.

[65] T. Kalker, G. Depovere, J. Haitsma, and M. Maes, "A video watermarking system for broadcast monitoring," in *Proc. SPIE IS&T/SPIE's 11th Annu. Symp., Electronic Imaging '99: Security and Watermarking of Multimedia Contents*, vol. 3657, Jan. 1999.

[66] M. S. Kankanhalli, Rajmohan, and K. R. Ramakrishnan, "Content-based watermarking of images," in *Proc. ACM Multimedia '98*, Bristol, U.K., Sept. 1998.

[67] M. Kobayashi, "Digital watermarking: Historical roots," IBM Research, Tokyo Res. Lab., Tech. Rep., Apr. 1997.

[68] E. Koch, J. Rindfrey, and J. Zhao, "Copyright protection for multimedia data," *Digital Media and Electronic Publishing*, 1996.

[69] E. Koch and J. Zhao, "Toward robust and hidden image copyright labeling," in *Proc. Workshop Nonlinear Signal and Image Processing*, Marmaros, Greece, June 1995.

[70] M. Kuhn. (1997, Nov.). StirMark. [Online]. Available WWW: http://www.cl.cam.ac.uk/~mgk25/stirmark/.

[71] D. Kundur and D. Hatzinakos, "A robust digital image watermarking method using wavelet-based fusion," in *Proc. IEEE Int. Conf. Image Processing 1997 (ICIP 97)*, vol. 1, Santa Barbara, CA, Oct. 1997, pp. 544–547.

[72] ——, "Digital watermarking using multiresolution wavelet decomposition," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing 1998 (ICASSP 98)*, vol. 5, Seattle, WA, May 1998, pp. 2969–2972.

[73] M. Kutter, F. Jordan, and F. Bossen, "Digital signature of color images using amplitude modulation," in *Proc. Electronic Imaging 1997 (EI 97)*, San Jose, CA, Feb. 1997.

[74] ——, "Digital signature of color images using amplitude modulation," *J. Electron. Imaging*, vol. 7, no. 2, pp. 326–332, Apr. 1998.

[75] M. Kutter and F. Petitcolas, "A fair benchmark for image watermarking systems," in *Proc. SPIE IS&T/SPIE's 11th Annu. Symp., Electronic Imaging '99: Security and Watermarking of Multimedia Contents*, vol. 3657, Jan. 1999.

[76] M. Kutter, "Watermarking resisting to translation, rotation, and scaling," in *Proc. SPIE Int. Symp. on Voice, Video, and Data Communication*, Nov. 1998.

[77] G. Langelaar, private communication.

[78] C. Langelaar, J. C. A. ven der Lubbe, and R. L. Lagendijk, "Robust labeling methods for copy protection of images," in *Proc. Electronic Imaging*, San Jose, CA, Feb. 1997, vol. 3022, pp. 298–309. [Online]. Available WWW: http://www-it.et.tudelft.nl/~gerhard/home.html.

[79] G. Langelaar, R. Lagendijk, and J. Biemond, "Watermarking by DCT coefficient removal: Statistical approach to optimal parameter settings," in *Proc. SPIE IS&T/SPIE's 11th Annu. Symp., Electronic Imaging '99: Security and Watermarking of Multimedia Contents*, vol. 3657, Jan. 1999.

[80] G. C. Langelaar, R. L. Lagendijk, and J. Biemond, "Real-time labeling methods for MPEG compressed video," in *Proc. 18th Symp. Information Theory in the Benelux*, Veldhoven, The Netherlands, May 1997.

[81] ——, "Removing spatial spread spectrum watermarks by non-linear filtering," in *Proc. Europ. Signal Processing Conf. (EUSIPCO '98)*, Rhodes, Greece, Sept. 1998.

[82] G. C. Langelaar, J. C. A. van der Lubbe, and J. Biemond, "Copy protection for multimedia data based on labeling techniques," in *Proc. 7th Symp. Information Theory in the Benelux*, Enschede, The Netherlands, May 1996. [Online]. Available WWW: http://www-it.et.tudelft.nl/~gerhard/home.html.

[83] J.-P. Linnartz. (1998). MPEG PTY marking. [Online] Available WWW: http://diva.eecs.berkeley.edu/linnartz/pty.html.

[84] S. Low and N. Maxemchuk, "Performance comparison of two text marking methods," *IEEE J. Select. Areas Commun. (Special Issue on Copyright and Privacy Protection)*, vol. 16, pp. 561–572, May 1998.

[85] S. Low, N. Maxemchuk, J. Brassil, and L. O'Gorman, "Document marking and identification using both line and word shifting," in *Proc. Infocom '95*, Boston, MA, Apr. 1995.

[86] H. D. Lüke, *Korrelationssignale* (in German). Berlin, Germany: Springer, 1992.

[87] B. Macq, J.-F. Delaigle, and C. De Vleeschouwer, "Digital watermarking," *SPIE Proc. 2659: Optical Security and Counterfeit Deterrence Techniques*, Mar. 1996, pp. 99–110.

[88] M. Maes, "Twin peaks: The histogram attack on fixed depth image watermarks" in *Lecture Notes in Computer Science*, vol. 1525. Berlin, Germany: Springer, 1998, pp. 290–305.

[89] M. J. J. J. B. Maes and C. W. A. M. Overveld, "Digital watermarking by geometric warping," in *Proc. Int. Conf. Image Processing (ICIP)*, vol. 1, Chicago, IL, 1998.

[90] K. Matsui and K. Tanaka, "Video-steganography," in *Proc. IMA Intellectual Property Project*, vol. 1, Jan. 1994, pp. 187–206.

[91] N. F. Maxemchuk and S. Low, "Marking text documents," in *Proc. IEEE Int. Conf. Image Processing 1997 (ICIP '97)*, vol. 3, Santa Barbara, CA, Oct. 1997, pp. 13–16.

[92] G. Nicchiotti and E. Ottaviani, "Non-invertible statistical wavelet watermarking," in *Proc. Europ. Signal Processing Conf. (EUSIPCO '98)*, Rhodes, Greece, Sept. 1998.

[93] N. Nikolaidis and I. Pitas, "Copyright protection of images using robust digital signatures," in *Proc. ICASSP '96*, Atlanta, GA, May 1996.

[94] R. Ohbuchi, H. Masuda, and M. Aono, "Embedding data in three-dimensional polygonal models," in *Proc. ACM Multimedia '97*, Seattle, WA, Nov. 1997.

[95] ——, "Watermarking three-dimensional polygonal models through geometric and topological modifications," *IEEE J. Select. Areas Commun. (Special Issue on Copyright and Privacy Protection)*, vol. 16, pp. 551–560, May 1998.

[96] J.-M. Chassery, P. Bas, and F. Davoine, "Self-similarity based image watermarking," in *Proc. Europ. Signal Processing Conf. (EUSIPCO '98)*, Rhodes, Greece, Sept. 1998.

[97] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*. New York: McGraw-Hill, 1991.

[98] A. Perrig, "A copyright protection environment for digital images," Diploma dissertation, École Polytechnique Fédérale de Lausanne, Switzerland, Feb. 1997.

[99] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Information hiding—A survey," this issue, pp. 1062–1078.

[100] ——, "Attacks on copyright marking systems," in *Lecture Notes in Computer Science: Information Hiding*. Berlin, Germany: Springer, 1998, pp. 218–238.

[101] R. Pickholtz, D. Schilling, and L. Milstein, "Theory of spread spectrum communications—A tutorial," *IEEE Trans. Commun.*, vol. COM-30, pp. 855–884, May 1982.

[102] R. L. Pickholz, D. L. Schilling, and L. B. Milstein, "Theory of spread-spectrum communications–A tutorial," *IEEE Trans. Commun.*, vol. 30, pp. 855–884, May 1982.

[103] I. Pitas, "A method for signature casting on digital images," in *Proc. Int. Conf. Image Processing (ICIP)*, Lausanne, Switzerland, Sept. 1996.

[104] I. Pitas and T. H. Kaskalis, "Applying signatures on digital images," in *Proc. IEEE Workshop Nonlinear Image and Signal Processing*, Neos Marmaros, Greece, June 1995, pp. 460–463.

[105] A. Piva, M. Barni, E. Bartolini, and V. Cappellini, "DCT-based watermarking recovering without resorting to the uncorrupted original image," in *Proc. IEEE Int. Conf. Image Processing (ICIP)*, vol. 1, Santa Barbara, CA, 1997, p. 520.

[106] C. Podilchuk and W. Zeng, "Watermarking of the JPEG bitstream," in *Proc. Int. Conf. Imaging Science, Systems, and Applications (CISST '97)*, Las Vegas, NV, June 1997, pp. 253–260.

[107] C. I. Podilchuk, "Digital image watermarking using visual models," in *Proc. Electronic Imaging*, vol. 3016, San Jose, CA, Feb. 1996.

[108] C. I. Podilchuk and W. Zeng, "Perceptual watermarking of still images," in *Proc. of Workshop Multimedia Signal Processing*, Princeton, NJ, June 1997.

[109] J. Puate and F. Jordan, "Using fractal compression scheme to embed a digital signature into an image," in *Proc. SPIE Photonics East'96 Symp.*, Boston, MA, Nov. 1996.

[110] L. Qiao and K. Nahrstedt, "Watermarking schemes and protocols for protecting rightful ownership and customer's rights," *J. Visual Commun. Image Representation*, vol. 9, no. 3, pp. 194–210, Sept. 1998.

[111] M. Ramkumar and A. Akansu, "On the choice of transforms for data hiding in compressed video," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing 1999 (ICASSP '99)*, Phoenix, AZ, Mar. 1999.

[112] J. J. K. Ó Ruanaidh, F. M. Boland, and O. Sinnen, "Watermarking digital images for copyright protection," in *Proc. Electronic Imaging and the Visual Arts*, Florence, Italy, Feb. 1996.

[113] J. J. K. Ó Ruanaidh, W. J. Dowling, and F. M. Boland, "Phase watermarking of digital images," in *Proc. Int. Conf. Image Processing (ICIP)*, vol. 3, Sept. 1996, pp. 239–242.

[114] J. J. K. Ó Ruanaidh and T. Pun, "Rotation, scale and translation invariant digital image watermarking," in *Proc. IEEE Int. Conf. Image Processing 1997 (ICIP 97)*, Santa Barbara, CA, vol. 1, Oct. 1997, pp. 536–539.

[115] ——, "Rotation, scale, and translation invariant spread spectrum digital image watermarking," *Signal Processing (Special Issue on Watermarking)*, vol. 66, no. 3, pp. 303–318, May 1998.

[116] K. Sayood, *Introduction to Data Compression*. New York: Morgan Kaufmann, 1996, ch. 13.

[117] B. Schneider, *Applied Cryptography*. New York: Wiley, 1996, ch. 4.

[118] T. Sikora, "Low complexity shape-adaptive [DCT] for coding of arbitrarily shaped image segments," *Image Commun. (Special Issue on Coding Techniques for Very Low Bit-Rate Video)*, vol. 7, nos. 4–6, Nov. 1995.

[119] M. K. Simon, J. K. Omura, R. A. Scholtz, and B. K. Levit, *Spread Spectrum Communications Handbook*. New York: McGraw-Hill, 1994.

[120] J. R. Smith and B. O. Comiskey, "Modulation and information hiding in images," in *Lecture Notes in Computer Science: Information Hiding*, vol. 1174. Berlin, Germany: Springer, 1996, pp. 207–226.

[121] H. S. Stone, "Analysis of attacks on image watermarks with randomized coefficients," NEC Res. Inst., Princeton, NJ, Tech. Rep., May 1996.

[122] J. K. Su and B. Girod, "On the imperceptibility and robustness of digital fingerprints," submitted for publication.

[123] ——, "Power spectrum condition for L2-efficient watermarking," submitted for publication.

[124] M. D. Swanson, M. Kobayashi, and A. H. Tewfik, "Multimedia data embedding and watermarking technologies," *Proc. IEEE (Special Issue on Multimedia Signal Processing)*, vol. 86, pp. 1064–1087, June 1998.

[125] M. Swanson, B. Zhu, and A. Tewfik, "Data hiding for video-in-video," in *Proc. IEEE Int. Conf. Image Processing 1997 (ICIP '97)*, vol. 2, Santa Barbara, CA, Oct. 1997, pp. 676–679.

[126] M. Swanson, B. Zhu, and A. Tewfik, "Multiresolution video watermarking using perceptual models and scene segmentation," in *Proc. IEEE Int. Conf. Image Processing 1997 (ICIP '97)*, vol. 2, Santa Barbara, CA, Oct. 1997, pp. 558–561.

[127] M. Swanson, B. Zhu, and A. H. Tewfik, "Multiresolution scene-based video watermarking using perceptual models," *IEEE J. Select. Areas Commun. (Special Issue on Copyright and Privacy Protection)*, vol. 16, pp. 540–550, May 1998.

[128] M. Swanson, B. Zhu, A. H. Tewfik, and L. Boney, "Robust audio watermarking using perceptual coding," *Signal Processing (Special Issue on Watermarking)*, vol. 66, no. 3, pp. 337–356, May 1998.

[129] M. D. Swanson, B. Zhu, and A. H. Tewfik, "Robust data hiding for images," in *Proc. IEEE Digital Signal Processing Workshop*, Loen, Norway, Sept. 1996, pp. 37–40.

[130] ——, "Transparent robust image watermarking," in *Proc. of Int. Conf. Image Processing (ICIP)*, Lausanne, Switzerland, Sept. 1996.

[131] K. Tanaka, Y. Nakamura, and K. Matsui, "Embedding secret information into a dithered multilevel image," in *Proc. 1990 IEEE Military Commun. Conf.*, Sept. 1990, pp. 216–220.

[132] ——, "Embedding the attribute information into a dithered image," *Syst. Comput. Japan*, vol. 21, no. 7, 1990.

[133] B. Tao and B. Dickinson, "Adaptive watermarking in the DCT domain," in *Proc. Int. Conf. Image Processing (ICIP)*, Lausanne, Switzerland, Sept. 1996.

[134] J. F. Tilki and A. A. Beex, "Encoding a hidden digital signature onto an audio signal using psychoacoustic masking," in *Proc. 7th Int. Conf. Digital Signal Processing Applications & Technology*, Boston, MA, Oct. 1996, pp. 476–480.

[135] A. Tirkel, private communication.

[136] A. Tirkel, G. Rankin, R. van Schyndel, W. Ho, N. Mee, and C. Osborne, "Electronic water mark," in *Proc. DICTA 1993*, Dec. 1993, pp. 666–672.

[137] A. Tirkel, R. van Schyndel, and C. Osborne, "A two-dimensional watermark," in *Proc. DICTA 1993*.

[138] (1997, July). UnZign watermark removal software. [Online]. Available WWW: http://altern.org/watermark/.

[139] R. G. van Schyndel, A. Z. Tirkel, and C. F. Osborne, "A digital watermark," in *Proc. Int. Conf. Image Processing (ICIP)*, vol. 2, 1994, pp. 86–89.

[140] A. J. Viterbi, *CDMA: Principles of Spread Spectrum Communication*. Reading, MA: Addison-Wesley, 1995.

[141] G. Voyatzis and I. Pitas, "Applications of toral automorphisms in image watermarking," in *Proc. Int. Conf. Image Processing (ICIP)*, vol. 3, Lausanne, Switzerland, Sept. 1996, pp. 237–240.

[142] ——, "Chaotic mixing of digital images and applications to watermarking," in *Proc. Europ. Conf. Multimedia Applications, Services, and Techniques (ECMAST)*, Louvain-la-Neuve, Belgium, May 1996.

[143] H. Wang and C. C. J. Kuo, "An integrated progressive image coding and watermark system," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing 1998 (ICASSP 98)*, vol. 6, Seattle, WA, May 1998, pp. 3721–3723.

[144] J. Weiner and K. Mirkes, *Watermarking* (no. 257 in Bibliographic Series). Appleton, WI: Inst. Paper Chemistry, 1972.

[145] T. Wiegand, M. Lightstone, D. Mukherjee, T. G. Campbell, and S. K. Mitra, "Rate-distortion optimized mode selection for very low bit rate video coding and the emerging H.263 standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 182–190, Apr. 1996.

[146] R. B. Wolfgang, C. I. Podilchuk, and E. J. Delp, "Perceptual watermarks for digital images and video," this issue, pp. 1108–1126.

[147] R. B. Wolfgang and E. J. Delp, "A watermark for digital images," in *Proc. Int. Conf. Image Processing (ICIP)*, Lausanne, Switzerland, Sept. 1996, pp. 219–222.

[148] ——, "A watermarking technique for digital imagery: Further studies," in *Proc. Imaging Science, Systems, and Technology*, Las Vegas, NV, June–July 1997, pp. 279–287.

[149] ——, "Overview of image security techniques with applications in multimedia systems," in *Proc. SPIE Int. Conf. Voice, Video, and Data Commun.*, Dallas, TX, Nov. 1997.

[150] M. Wu, M. L. Miller, J. A. Bloom, and I. J Cox, "A rotation, scale, and translation resilient public watermark," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing 1999 (ICASSP '99)*, Phoenix, AZ, 1999.

[151] X. Xia, C. Boncelet, and G. Arce, "A multiresolution watermark for digital images," in *Proc. IEEE Int. Conf. Image Processing 1997 (ICIP '97)*, vol. 1, Santa Barbara, CA, Oct. 1997, pp. 548–551.

[152] W. Zhu, Z. Xiong, and Y.-Q. Zhang, "Multiresolution watermarking for images and video: a unified approach," in *Proc. Int. Conf. on Image Processing (ICIP)*, Chicago, IL, 1998.

**Frank Hartung** (Student Member, IEEE) received the M.Sc. degree in electrical engineering from the Technical University of Aachen, Germany. He was a Ph.D. student at the Telecommunication Lab of the University of Erlangen–Nuremberg, Erlangen, Germany, where he worked on video watermarking and video compression.

Since the spring of 1999, he has been with the Research Department of Ericsson Eurolab, Herzogenrath, Germany, working on multimedia. His research interests include digital watermarking of video and other multimedia data, video compression and transmission, multimedia systems and technology, and telecommunications technology.

**Martin Kutter** received the B.Sc. degree from the Technikum Winterthur Ingenieurschule, Switzerland, in 1989 and the M.Sc. degree in electrical engineering from the University of Rhode Island, Kingston, in 1996. He is currently pursuing the Ph.D. degree at the Signal Processing Laboratory, Swiss Federal Institute of Technology, Lausanne, Switzerland.

From 1992 to 1994, his was working in the R&D department of a company in the medical industry. His research interests include digital watermarking, cryptography, data compression, and image morphing.

## This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

## BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

☐ BLACK BORDERS

☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES

☐ FADED TEXT OR DRAWING

☑ BLURRED OR ILLEGIBLE TEXT OR DRAWING

☐ SKEWED/SLANTED IMAGES

☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS

☐ GRAY SCALE DOCUMENTS

☐ LINES OR MARKS ON ORIGINAL DOCUMENT

☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY

☐ OTHER: ____________________

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**

XP 000577034

P.D. 5.3.95
p. 20-25 (6)

# NetBill: An Internet Commerce System Optimized for Network Delivered Services

Marvin Sirbu
Engineering and Public Policy Dept.

J. D. Tygar
Computer Science Dept.

Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

## Abstract

*Netbill is a business model, set of protocols, and software implementation for supporting commerce in information goods and other network delivered services. It has very low transaction costs for micropayments (around 1¢ for a 10¢ item), protects the privacy of the transaction, and is highly scalable. Of special interest is our new certified delivery mechanism which delivers information goods if and only if the customer has payed for them. This paper discusses the design of the NetBill protocol and our World Wide Web (WWW) prototype implementation*

## Introduction

As the explosive growth of the Internet continues, more people rely on networks for timely information. However, since most information on the Internet today is free, intellectual property owners have little incentive to make valuable information accessible through the network. There are many potential providers who could sell information on the Internet and many potential customers for that information. What is missing is an electronic commerce mechanism that links the merchants and the customers.

NetBill is a business model, set of protocols, and software implementation allowing customers to pay owners and retailers of information. While NetBill will enable a market economy in information, we still expect that there will be an active exchange of free information.

## The market for information

Porat and others have shown that information industries dominate the economy [1]. Estimates of the market for on-line information vary from $10 billion to $100 billion per year depending upon how the market is defined [2]. There are more than 15,000 databases accessible over networks. Vendors can distribute information products varying from complex software valued at thousands of dollars per copy, to journal pages or stock quotes valued at a few pennies each. A challenge for network-based electronic commerce is to keep transaction costs to a small fraction of the cost of the item. The desire to support *micropayments* worth only a few pennies each is a driving factor in the NetBill design.

A second challenge in the information marketplace is supporting *micromerchants*, who may be individuals who sell relatively small volumes of information. Merchants need a simple way of doing business with customers over networks, so that the costs of setting up accounting and billing procedures are minimal. A model for micromerchants is the French Minitel system, which provides 20,000 "kiosks" offering computer-based services to Minitel users. Many of these kiosks are provided by small entrepreneurs who enter the marketplace for little more than the cost of a PC and the labor to acquire or develop valuable information.

The purchase of goods over a network requires linking two transfers: the transfer of the goods from the merchant to the customer, and the transfer of money from the customer to the merchant. In the case of physical goods, a customer can order the goods and transfer money over the network, but the goods cannot be delivered over the network. Information goods have the special characteristic that both the delivery of the goods *and* the transfer of money can be accomplished on the same network. This allows for optimizations in the design of an electronic commerce system.

## A NetBill scenario

Figure 1 shows NetBill's model. A user, represented by a client computer, wishes to buy information from a merchant's server. A NetBill server maintains accounts for both customers and merchants. These accounts are linked to conventional financial institutions. A NetBill transaction transfers the information goods from merchant to user, and debits the customer's account and credits the merchant's account for the value of the goods. When necessary, funds in a customer's NetBill account can be replenished from a bank or credit card; similarly funds in a merchant's NetBill account are made available by depositing them in the merchant's bank account.

The transfer of an information good consists of delivering bits to the customer. This bit sequence may have any internal structure, for example, the results of a database search, a page of text, or a software program. Users may be charged on a per item basis, or by a subscription allowing unlimited access, or by a number of other pricing models.

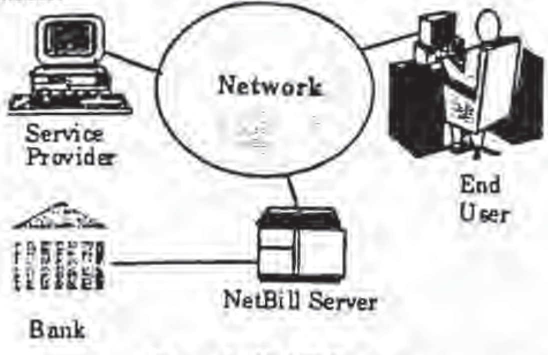


*Figure 1: Netbill Concept*

Once the customer receives the bits, there are no technical means to absolutely control what the customer does with them. For example, suppose an information provider wants to charges different price for pages viewed on-line, versus printed pages. The merchant can provide customers with client software distinguishing viewing from printing, and which initiates a new billing transaction when the screen is printed. However, there are no technical means to prevent the user from tampering with that software once it is on her machine; a corrupt user who has only paid to view the bits could thus bypass the charge for printing. Merchants may still choose to distribute special software in the belief that tampering will be infrequent. Similarly, there is no technical means to prevent users from violating copyright by redistributing information [3].

## NetBill design

There are a number of challenges to making electronic commerce systems feasible:

- *High transaction volumes at low cost.* If information is sold for a few pennies a page, then an electronic commerce system must handle very large transaction volumes at a marginal cost of a penny or less per transaction.
- *Authentication, privacy and security.* The Internet today provides no universally accepted means for authenticating users, protecting privacy, or providing security.
- *Account management and administration.* Users and merchants must be able to establish and monitor their accounts.

This paper describes the architecture of NetBill, a system designed to meet these goals. Our students and we have implemented three generations of NetBill prototypes. We hope to soon mount a trial in which various forms of information are sold to users using NetBill.

## NetBill architecture

NetBill uses a single protocol that supports charging in a wide range of service interactions. NetBill provides transaction support through libraries integrated with different client-server pairs. These libraries use a single transaction-oriented protocol for communication between client and server and NetBill; the normal communications model between client and server is unchanged. Clients and servers can continue to communicate using protocols optimized for the application — for example, video delivery or database queries — while the financial-related information is transmitted over protocols optimized for that purpose. This approach allows NetBill to work with information delivery mechanisms ranging from the WWW to FTP and MPEG-2 streams.

The client library — which we call the *checkbook* — and the server library — the *till* — have a well-defined API allowing easy integration with a range of applications. (Below we describe how we integrated these libraries with Mosaic clients and HTTP servers.) The libraries incorporate all security and payment protocols, relieving the client/server application developer from having to worry about these issues. All network communications between the checkbook and till are encrypted to protect against adversaries who eavesdrop or inject messages.

## The NetBill transaction protocol

Before a customer begins a typical NetBill transaction, she will usually contact a server to locate information or a service of interest. For example, the customer may request a Table of Contents of a journal showing available articles available, and a list price associated with each article. The transaction begins when the customer requests a formal price quote for a product. This price may be different than the standard list price because, for example, the customer may be part of a site license group, and thus be entitled to a marginal price of zero [4]. Alternatively, the customer may be entitled to some form of volume discount, or perhaps there is a surcharge during the peak hour.

Requesting the price quote is easy. As we discuss below, in a WWW browser application we have built, a customer requests a price quote by simply clicking on a displayed article reference.

The customer's client application then indicates to the checkbook library that it would like a price quote from a particular merchant for a specified product. The checkbook library sends an authenticated request for a quote to the till library which forwards it to the merchant's application. (Figure 2, Step 1.)

The merchant then must invoke an algorithm to determine a price for the authenticated user [5]. He returns the digitally signed price quote through the till, to the checkbook (Step 2), and on to the customer's application. The customer's application then must make a purchase decision. The application can present the price quote to the customer or it can approve the purchase without prompting the customer. For example, the customer may specify that her client software accept any price quote below some threshold amount; this relieves her of the burden of assenting to every low-value price quotes via a dialog box.

Assume the customer's application accepts the price quote. The checkbook then sends (Step 3) a digitally signed purchase request to the merchant's till. The till then requests the information goods from the merchant's application and sends them to the customer's checkbook encrypted in a one-time key (Step 4), and computes a cryptographic checksum (such as MD5 [6]) on the encrypted message. As the checkbook receives the bits, it writes them to stable storage. When the transfer is complete, the checkbook computes its own cryptographic checksum on the encrypted goods and returns to the till a digitally signed message specifying the product identifier, the accepted price, the cryptographic checksum, and a timeout stamp: we refer to this information as the *electronic payment order (EPO)* (Step 5). Note that, at this point, the customer can not decrypt the goods; neither has the customer been charged.

Upon receipt of the EPO, the till checks its checksum against the one computed by the checkbook. If they do not match, then the goods can either be retransmitted, or the transaction aborted at this point. This step provides very high assurance that the encrypted goods were received without error.

If checksums match, the merchant's application creates a digitally signed invoice consisting of price quote, checksum, and the decryption key for the goods. The application sends both the EPO and the invoice to the NetBill server (Step 6).

The NetBill server verifies that the product identifiers, prices and checksums are all in agreement. If the customer has the necessary funds or credit in her account, the NetBill server debits the customer's account and credits the merchant's account, logs the transaction, and saves a copy of the decryption key. The NetBill server then returns to the merchant a digitally signed message containing an approval, or an error code indicating why the transaction failed (Step 7). The merchant's application forwards the NetBill server's reply and (if appropriate) the decryption key to the checkbook (Step 8).

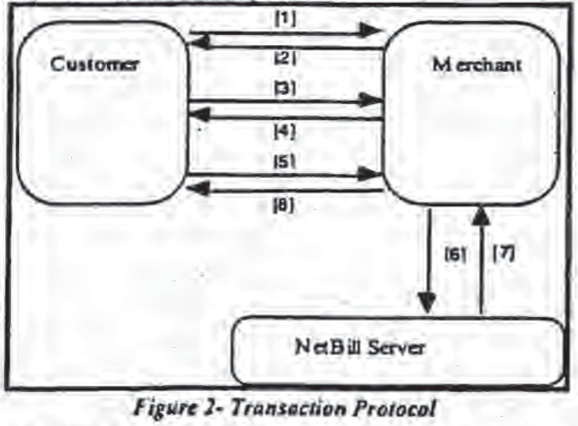


*Figure 2- Transaction Protocol*

*Protocol Failure Analysis*

The above description assumed that no failures occurred during the execution of the protocol. In reality, the protocol must gracefully cope with network and host failures. One of our goals is to tightly link two events: charging the customer and delivering the goods. The customer should pay exactly when she receives the information goods.

The NetBill server is highly reliable and highly available. All transactions at the NetBill server are atomic: they either finish completely or not at all. NetBill is never in doubt about the status of a purchase. We cannot make similar assumptions about the reliability of the merchant's and customer's software; they must maintain a state consistent with the NetBill Server.

First, consider the protocol from the perspective of the customer's application. Up to step 5, when the customer application acknowledges receipt of the information goods, the customer application knows that no transaction has occurred. That is, the customer does not have access to the product and the merchant does not have the customer's money. Once the application sends the EPO, the customer is *committed* to the transaction and must be prepared to accept the purchase. If the customer's application does not receive a response from the merchant's application, then it is the responsibility of the customer's application to determine what happened: the customer's application can poll either the merchant application or the NetBill server to determine the status of the purchase request. If the merchant's application did not successfully forward the EPO to the NetBill server, then the EPO will have expired and the NetBill server will respond to the customer's application that the purchase has failed. Of course, the customer still does not have the one time key, so while the customer still has her money, she also does not have the goods. If, on the other hand, the transaction succeeded before communication failed, then the customer's application can find the status of the purchase and, if appropriate, the decryption key from either the merchant's application or the NetBill server (which has registered the key). If both are unreachable, the customer's application must continue to poll.

Now consider the protocol from the perspective of the merchant's application. Before it forwards the EPO and invoice to the NetBill server, the merchant's application knows that the transaction has not occurred. After it forwards the EPO and invoice, however, the merchant's application is *committed* to the transaction and must obtain the result from the NetBill. If the merchant's application does not receive a response from the NetBill server, the merchant's application must poll the NetBill server.

The protocol is much simpler for the NetBill server than for the other parties. The NetBill server is never in a state in which it depends on a response from another entity to determine the status of a transaction. Until the NetBill server receives the EPO and invoice from the merchant's application, it knows nothing about the purchase. Once it receives the EPO and invoice it has all the information necessary to approve or reject the purchase.

We use the term ***certified delivery*** to describe the mechanism of delivering encrypted information goods and then charging against the customer's NetBill account, with decryption key registration both at the merchant's application and the NetBill server.

The NetBill transaction protocol also exhibits a number of other desirable features:

- *Support for flexible pricing.* By including the steps of offer and acceptance, we provide an opportunity for the merchant to calculate a customized quote for an individual customer. In the process we also generate signed messages that can later prove that there was a contract at the quoted price.
- *Scalability.* The bottleneck in the NetBill model is the NetBill server which supports many different merchants. Our transaction protocol minimizes the load on the NetBill server and distributes the burden over the many customer and merchant machines. Note that a single interaction with the NetBill server both verifies the availability of funds and records the transaction. It is not possible to have less than one interaction with the NetBill server [7].
- *Protection of user accounts* against unscrupulous merchants. In a conventional credit card transaction, the merchant learns the customer's credit card number and can submit fraudulent invoices in the customer's name. In a NetBill transaction, the customer digitally signs the EPO using a key that is never revealed to the merchant, thus eliminating this threat. Moreover, the customer has proof of the exact nature of the information goods received, providing evidence in case a dishonest merchant attempts to deliver faulty information goods.

## NetBill account management

In this section, we discuss how customers and merchants can manage their NetBill accounts.

NetBill supports a many-to-many relationship between *customers* and *accounts*. A project account at a corporation can have many users authorized to charge against it. Conversely, an individual customer can maintain multiple personal accounts. Every account has a single user who is the account *owner*; and the account owner can grant various forms of access rights on the account to other users.

User account administration is provided through WWW forms. Using a standard WWW browser, an authorized user can view and change a NetBill account profile, authorize funds transfer into that account, or view a current statement of transactions on that account. Authentication and security are provided by treating account information as "billable" items. NetBill provides account information to users using the NetBill protocol. NetBill can be configured to provide this information for free or for a service charge, as desired.

Automating account establishment for both customers and merchants is important for limiting costs. (Account creation is one of the largest costs associated with traditional credit card and bank accounts.) To begin the process, a customer retrieves, perhaps by anonymous FTP, a digitally signed NetBill security module that will work with the user's WWW browser. Once the customer checks the validity of the security module, she puts the module in place. She then fills out a WWW form, including appropriate credit card or bank account information to fund the account, and submits it for processing. The security module encrypts this information to protect it from being observed in transit. The NetBill server must verify that this credit card or banking account number is valid and that the user has the right to access it. There are a variety of techniques for this verification: for example, customers may telephone an automated attendant system and provide a PIN associated with the credit card or bank account to obtain a password.

## NetBill costs and interaction with financial institutions

In a modern market economy, there are many forms of money, but two distinct poles typify the range of alternatives: *tokens* and *notational money*. Currency consists of unforgeable tokens that are widely accepted by both buyers and sellers as a store of value. In a cash transaction, the seller delivers goods to the customer while the customer delivers currency to the seller. Other projects are developing forms of electronic currency for network commerce based on unique digital bit strings.[8]

Demand deposit accounts at a bank are an example of notational money: on instruction (a check) by a customer, funds move from one ledger to another. A complex system involving intermediaries such as the Federal Reserve supports check clearing and settlements when the accounts are held at different banking institutions. Settlements can involve significant delays during which funds are not available to either party in a transaction. Notational accounts can have either a positive or negative balance, depending upon whether a bank is willing to extend credit to a buyer. For example, a credit card account runs a negative balance as the issuing bank executes instructions to transfer funds to a merchant's bank account.

Orders to transfer notational money are increasingly sent using electronic mechanisms: FedWire, automated clearinghouses (ACH), credit card authorization and settlement networks, and automated teller machine networks are all examples. NetBill also uses notational money. Because both customers and merchants maintain NetBill accounts, inter-institutional clearing costs are not incurred for every transaction. NetBill accounts provide a low cost mechanism to aggregate small value transactions before invoking a relatively high fixed cost conventional transaction mechanism. Customers move money into their NetBill account in large chunks (for example, $50 - $100) by charging a credit card or through an ACH transaction. Similarly, money moves from a merchant's NetBill account to the merchant's bank through an ACH deposit transaction.

NetBill accounts can be either pre-paid (debit model) or post-paid (credit model). In the prepaid model, funds would be transferred to NetBill in advance to cover future purchases. If the user does not have sufficient funds to cover a particular transaction, that transaction would be declined. The amount of any prepayment is set by the customer, subject to minimums and maximums established by the NetBill operator. On pre-paid accounts, the system allows users to designate the balance at which she is prompted to transfer additional funds to Netbill. Because ACH transactions take several days to clear, a user prepaying her Netbill account through the ACH may not have immediate access to the funds. Funding through a credit card, while incurring larger transaction fees, allows immediate access to a prepayment.

In the credit model, transactions would be accumulated with payment to NetBill being triggered by either time (based on a pre-established billing period) or dollar amount (based on a pre-established limit). Because granting credit creates a risk of non-payment, higher transaction fees may be associated with credit, versus prepaid accounts.

The design space for electronic transaction systems has three crucial dimensions: risk, delay and cost. For immediate transactions, risks of fraud or non-payment can be dealt with in two ways: 1) incorporating an insurance fee proportional to the transaction amount, or 2) investing in sophisticated security systems with (high) fixed costs independent of transaction size. Credit card systems are of the first type, typically charging 1-3% of the value of the transaction, while FedWire takes the second approach. Delay can reduce risk by allowing verification of fund availability before committing a transaction, and by allowing batching to achieve economies of scale, particularly in interbank settlements. However, delay imposes opportunity costs when funds are not available until cleared.

NetBill is optimized for very low marginal transaction costs (on the order of 1¢) on small value transactions (on the order of 10¢.) Fixed networking costs are reduced by using the Internet with its substantial economies of scale, as opposed to a dedicated single function network. Because both customers and merchants maintain accounts at NetBill, most transfers are internal to NetBill; this reduces both risk and processing cost. When fund transfers outside NetBill' are necessary, they can take advantage of aggregation, which spreads fixed transaction costs over larger sums. Use of ACH transfers and prepaid accounts minimizes risk at the cost of some delay before incoming funds are available; where NetBill offers deposits through credit cards, or grants credit itself, the risk increases and must be passed on to customers as higher fees.

NetBill keeps other costs of operation low by: automating all account administration functions; using techniques like certified delivery to reduce the incidence of complaints and customer service costs; and using a modern distributed processing approach for the core NetBill processing system.

## An example of NetBill with Mosaic

Because WWW browsers and servers are a *de facto* standard for distributing information over the Internet, we have created a prototype implementation of NetBill that allows for billing of WWW transactions. Rather than link the NetBill libraries with a WWW browser and http server respectively, we have enabled commerce with no modification to either the browser or the server. Our design introduces two entities in order to support the exchange of money for goods: the Money Tool and the Product Server. The Money Tool runs on the customer's machine and works with a Mosaic browser. It allows the customer to authenticate, select accounts, approve/deny transactions, and monitor expenditures. The Product Server, which incorporates the till libraries, works with the http server to sell information products.

When a user clicks on a product in a product server's catalog, the server returns a special file with a mime type containing information about the server's identity, the product to be ordered, and the port number of the product server. This mime type spawns a "helper" program in the same way that jpeg, sound, and mpeg files currently do. The spawned program communicates the contents of the file between Mosaic and the Money Tool.

The Money Tool acts as the customer's application in the NetBill transaction protocol described above. After it receives and decrypts the goods, it uses the remote control function of Mosaic to cause the browser to display the received information. Besides implementing the steps in the protocol, the Money Tool provides a number of useful functions to help the user manage transactions (Figure 3):

Session begun on: Mon Dec 19 12:35:21 1994
Amount spent this session: $0.20
Balance: $4903.42
(For account #67892 as of Mon Dec 19 12:38:13 1994)
Current Account: Market Research (#67892)

Registry of Remitted Checks

Mon Dec 19 12:37:41 1994 CheckNo #3
Account: #67892, Amount: $0.20, Product ID: #510
Made to: m75@NETBILL.INI.CMU.EDU Memo: "elec. comm"

Total checks: 1
Total amount of checks: $0.20

*Figure 3: The Money Tool*

- it provides an authentication dialog window
- it provides a running total of expenditures in the current session and the current balance in the user's NetBill account
- it provides a listing of all EPOs processed in the current session
- it can be configured to automatically approve expenditures below a threshold
- it can be used to retrieve the product encryption key from NetBill in the event of failure of a merchant host.

Spyglass has recently proposed a standard API for Security Plug-in Modules for WWW browsers [9]. In the future we expect to integrate the Money Tool with the browser using this mechanism.

In the current implementation, the initial request for goods to the http server causes the server to run a script that writes information about the request to a temporary file at the server. When the Product Server receives a request for a price quote from the Money Tool, it must access the server's database to determine the price quote based on the customer identity. If the quotation is approved, the product server finds the goods using the information saved by the http server and completes the NetBill transaction protocol.

## Additional Issues

As described above, NetBill is well suited for supporting commerce in information goods. However, the NetBill model can also be extended in a variety of ways to support other types of purchases. For example, NetBill could be used equally well for conventional bill paying. A customer could view a bill presented as a Web page; instead of buying information goods, we can think of the customer as buying a receipt for having paid the bill.

If the product to be bought is a one hour movie, it is likely that the customer will want to stream the data directly to a viewer, which conflicts with NetBill's model of certified delivery. We are exploring alternative approaches such as using the standard NetBill protocol to periodically buy a key for the next N minutes of an encrypted video stream.

We are also exploring the software rental application. A software vendor could incorporate the checkbook library in any arbitrary application software. Periodically, the software would ask the user to approve the purchase of a key for the next month's operation. (This requires mechanisms to prevent the software vendor from including a Trojan Horse designed to capture a renter's password.)

## Acknowledgments

Much of the development of NetBill has been done by students in project courses taken as part of Carnegie Mellon's graduate program in Information Networking. We thank all of those students for their help and ideas. Support for our research was provided in part by a grant from the National Science Foundation.

## Notes

For more information on NetBill, including a fuller version of this paper, please look at our WWW page at http://www.ini.cmu.edu:80/netbill.

1. Porat, M., *The Information Economy* (US. Office of Telecommunications, 1977)
2. *New York Times*, June 7, 1992
3. Separately, we are researching means of embedding a unique watermark in each copy sold which would allow illegal copies to be traced to the source.
4. In the special case of free information, we can optimize our protocol still further.
5. In separate work, we are designing pricing servers that can handle a very broad range of pricing strategies.
6. Rivest, *The MD5 Message Digest Algorithm*, April 1992.
7. In theory, one might bundle several transactions together and have them all processed as part of one interaction with NetBill. However usage data collected from Carnegie Mellon's Library Information System indicates that in the majority of cases, users contacting the library are looking for a single item, suggesting that bundling would not be appropriate. Cf. O'Toole, K., *The Internet Billing Server: Transaction Protocol Alternatives*, Carnegie Mellon Information Networking Institute Technical Report TR 1994-1.
8. Chaum, D., "Achieving electronic privacy", Scientific American, 267, No. 2, pp. 76-81, 1992
9. Jeff Hostetler, "A Framework for Security," 2nd WWW Conference, Chicago, Illinois, October, 1994.

XP004138681

Computer Networks and ISDN Systems 30 (1998) 1501–1510 ⩧ (10)

COMPUTER
NETWORKS
and
ISDN SYSTEMS

p d · 09/'98

# A status report on the *SEMPER* framework for secure electronic commerce

E

Matthias Schunter [a,*], Michael Waidner [b], Dale Whinnett [c]

[a] *Universität Dortmund, Informatik 6, D-44221 Dortmund, Germany*
[b] *IBM Zurich Research Laboratory, Säumerstraße 4, CH-8803 Rüschlikon, Switzerland*
[c] *Universität Freiburg, Friedrichstraße 50, D-79098 Freiburg, Germany*

## Abstract

The goal of the ACTS Project *SEMPER* (Secure Electronic Marketplace for Europe) is to provide the first open and comprehensive solution for secure commerce over the Internet and other public information networks. The basic framework described in an earlier article [M. Schunter, M. Waidner, Architecture and design of a secure electronic marketplace, Joint European Networking Conference (JENC8), Edinburgh, June 1997, pp. 712.1–712.5] has now been implemented in the Java programming language. It includes the payment systems SET, Chipper, and ecash™. The prototype uses a distinguished user-interface for trustworthy user in- and output which enables to use *SEMPER* on secure hardware. This article describes recent refinements to the *SEMPER* Framework as well as experiences gained in the field trials of the *SEMPER* software. © 1998 Elsevier Science B.V. All rights reserved.

*Keywords:* Electronic commerce; Framework; *SEMPER*; Fair exchange

## 1. Introduction

A wide range of businesses are rapidly moving to explore the huge potential of networked information systems, especially with the Internet-based WWW (World-Wide Web). Although the Internet has its roots in academia and is still dominated by free-of-charge information, dramatic changes are expected in the near future.

The goal of the 9-million ECU project, *SEMPER* (Secure Electronic Marketplace for Europe) [5,6], is to provide the first open and comprehensive solution for secure commerce over the Internet and other public information networks.

The members of the *SEMPER* consortium are *Commerzbank* (D), *Cryptomathic* (DK), *DigiCash* (NL), *EUROCOM EXPERTISE* (GR), *Europay International* (B), *FOGRA Forschungsgesellschaft Druck* (D), *GMD – German National Research Center for Information Technology* (D), *IBM* (CH, F), *INTRACOM* (GR), *KPN Research* (NL), *Maris* (NL), *Otto-Versand* (D), *r³ security engineering* (CH), *CNET* (F), *SINTEF* (N), *SSL* (UK), *Stichting Mathematisch Centrum/CWI* (NL), *Universities of Dortmund, Freiburg*, and *Saarbrücken* (D). Sponsoring partners are *Banksys* (B), *Banque Generale du Luxembourg* (LU), and *Telekurs* (CH). IBM Zurich Research Laboratory provides the technical leadership for the project.

---

[*] Corresponding author. E-mail: schunter@acm.org.

PII: S0169-7552(98)00186-X

### *1.1. Roles and services in the marketplace*

Like in a physical marketplace, the main purpose of an electronic marketplace is to bring potential *sellers* and *buyers* together:

- Sellers *offer* their goods and buyers *order* these goods; together this is a two-party *negotiation*, sometimes ending with an *agreement*.
- Sellers *deliver* their goods and buyers make *payments*; together this is a two-party *(fair) exchange*.
- Buyers or sellers might be dissatisfied with what has happened so far, i.e., several *exception handlers* and *dispute handlers* which may involve an *arbiter* are necessary.

In all these actions, the parties have specific *security requirements*, namely integrity, confidentiality, and availability. Confidentiality includes anonymity, which is often a requirement for browsing catalogues or for low-value purchases. Examples of typical scenarios of electronic commerce are:

- *Mail-order Retailing*: A retailer accepts electronic orders and payments, based on digital or conventional catalogues, and delivers physical goods.
- *On-line Purchase* of Information and Subscriptions: Like mail-order retailing, but with digital, maybe copyright-protected goods that are delivered on-line.
- *Electronic Mall*: An organisation offers services for several service providers, ranging from directory services ('index') over content hosting to billing services.
- *Contract Signing*: Two or more parties exchange signed copies of the same *statement*.

Naturally, an open system for electronic commerce cannot be restricted to these scenarios. It should be easily configurable and extensible to a broad range of different scenarios.

### *1.2. What is new in SEMPER?*

*SEMPER* is the first project that aims at the *complete* picture of secure electronic commerce, not just on specific pieces (like electronic payments), specific scenarios (like electronic on-line purchases) or specific products and protocols (an overview can be found at http://www.semper.org/sirene/outsideworld/ecommerce.html).

*SEMPER* provides an open framework which enables the integration of any protocol and product providing the necessary services. Therefore, applications are not restricted to specific proprietary technology or specific protocols.

Special attention is paid to customer anonymity and privacy. *SEMPER* develops an integrated anonymity management scheme extending the existing concepts for anonymous communication and credentials.

### *1.3. Recent changes*

Compared to an earlier description of *SEMPER* [4], this revision describes recent changes to the architecture such as a new framework for fair exchange as well as new anonymity services. Furthermore, it includes an extensive description of the *SEMPER* trials.

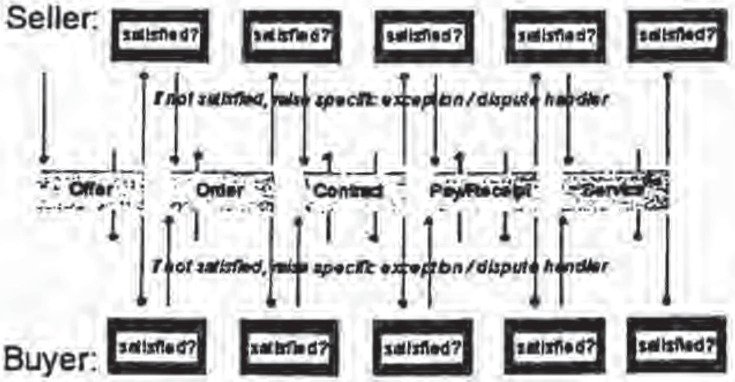


Fig. 1. Electronic commerce is a sequence of transfers and exchanges. Note that the protocol might enable other sequences as well, e.g., after 'Contract', 'Payment without Receipt' might also be enabled.

## 2. Model for electronic commerce

The framework described in this paper is based on a generic model for two-party electronic commerce. This model describes the flow of control as well as actions, and decisions for any commerce service. The main idea of the model for electronic commerce is describing business scenarios in terms of sequences of *transfers* and *exchanges* of data with decisions based on the success of these actions (see Fig. 1). This model is similar to the *dialogues* of interactive EDI.

### 2.1. Atomic actions: exchanges

The interactive actions between two players are *transfers* and *exchanges*. In a *transfer*, one party sends a package of business items to one or more other parties. The sending party can define certain security requirements, such as confidentiality, anonymity, or non-repudiation of origin.

A *fair exchange* is a simultaneous exchange of packages of business items among two parties. The parties have the *assurance* that their packages are sent if and only if the peer entity send their package as expected. Either both packages are exchanged or none. If no fairness guarantee is required, we can model such an exchange by two transfers. Business items which can be exchanged include

- *credentials*, such as access rights,
- *statements*, such as signed documents, certificates, or program and video data, and
- *money*, such as credit-card, cash, or bank transfer payments.

Fig. 2 gives an overview of the possible exchanges of these primitive types. Transfers are included as exchanges of 'something' for 'nothing'.

### 2.2. Electronic commerce: sequence of exchanges

The transfers and exchanges are fixed in our model given the data types and security attributes. Any business scenario is modelled as a sequence of exchanges with user-interaction and local decisions between successive exchanges (see Fig. 1).

In the course of an ongoing business, after each transfer or exchange, the parties are either

- *satisfied*, and thus willing to proceed with a certain number of other transfers or exchanges, or
- *dissatisfied*, in which case an *exception* or *dispute* is raised which might end up at a real court if all else fails,

depending on the success of the previous exchange, the items received, and possibly user-input. After each round, a decision as to whether and how to proceed is made.

## 3. The *SEMPER* framework

The *SEMPER* framework (Fig. 3) is structured in layers. The lowest layer deals with low-level security primitives and other *supporting services*,

| Transfer / Exchange *of* → *for* ↓ | Money | Credential | Information |
|---|---|---|---|
| Nothing (i.e., Transfer) | Payment | Certificate transfer etc. | Information transfer |
| Money | Fair money exchange | Fair payment with receipt | Fair purchase |
| Credential | ▨ | Fair contract signing | Fair conditional access |
| Information | ▨ | ▨ | Fair information exchange |

Fig. 2. Transfers and exchanges of primitive types.

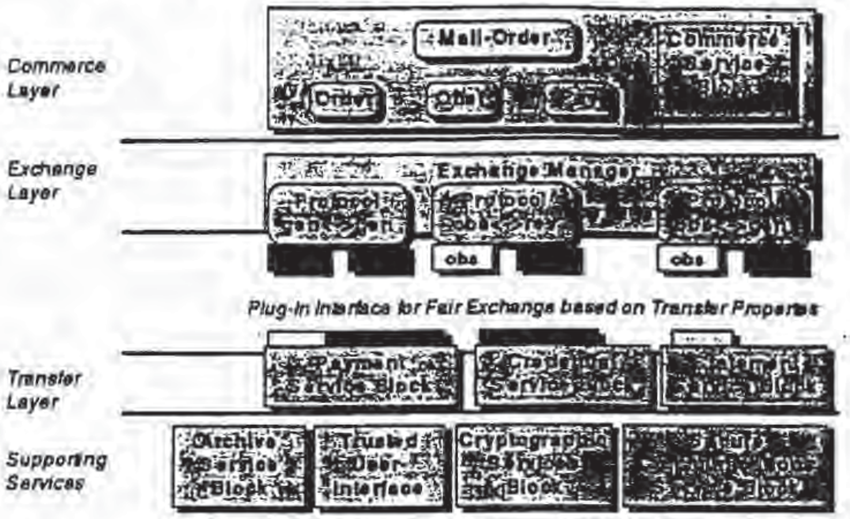


Fig. 3. Revised electronic commerce framework of *SEMPER*.

whereas the highest layer deals with commerce issues only:

- The *supporting services* are the usual cryptographic services, communication, archiving of data (keys, non-repudiation tokens, audit trail), setting preferences, access control and the trusted user interface. Furthermore, it provides secure communication services implementing security services such as anonymity which must be guaranteed for multiple atomic actions in one commerce session.
- The *transfer layer* provides services for transferring and grouping business items. This includes transfer-related security services such as non-repudiation of origin.
- The *exchange layer* supports fair exchange of business items, i.e., both participants input some business items and a description of the items expected in exchange. Fairness for exchanges means that the participants send their items if and only if they receive a transfer of the expected items. It implements the fair exchanges of the electronic commerce model such as contract signing, fair purchase, or non-repudiation of receipt.
- The *commerce layer* offers high-level services for business scenarios like 'mail-order retailing', 'online purchase of information', or 'registration with service provider'. It is configurable by downloading new services or extending existing ones.

### *3.1. Integration into the World-Wide Web*

The integration of the Framework into the World-Wide Web is depicted in Fig. 4. The integration code does not perform any security services but rather enables integration of *SEMPER* into different environments. It may use existing protocols such as 'cgi' or 'http' for this integration.

### *3.2. Commerce services*

The commerce layer implements the flow of control of our model using the transfer and exchange service for interactions with the peer, and the supporting services for user-interaction and persistent storage. It also performs the trust management and access control necessary for downloading certified commerce services.

The *Commerce Layer* provides services that directly implement protocols of business scenarios, e.g., how specific merchants or types of merchants handle customer registration and offering, ordering, payment, and delivery of goods. It implements the flow of control, i.e., the enabled sequences of exchanges, of the electronic commerce model. A set of client and server commerce services is the electronic equivalent of the 'terms of business' for the seller. The commerce layer does not only offer entire commerce protocols, but also building blocks

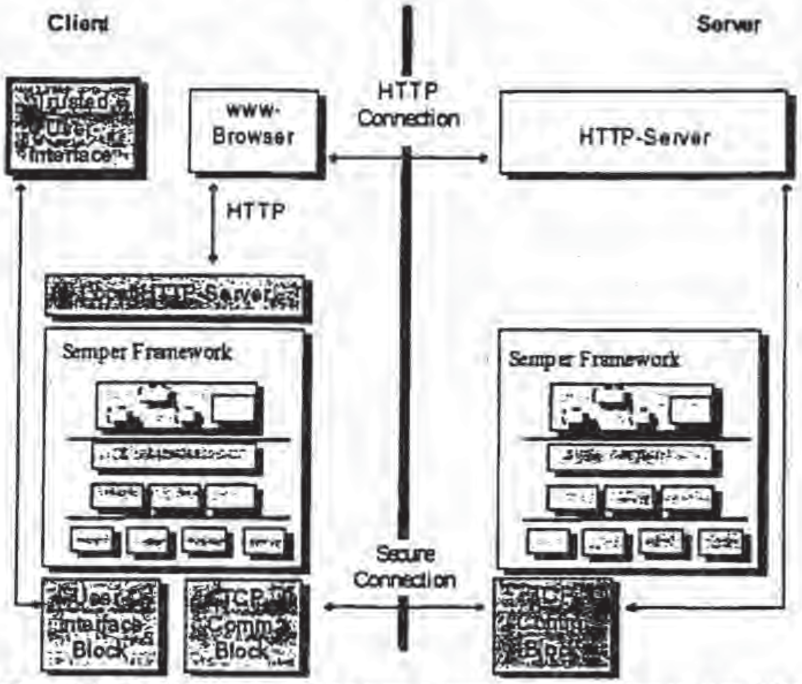


Fig. 4. Client- and server-side integration of the electronic commerce framework into the World-Wide Web.

that may be of more general use, in particular services to manage and fill out standardized order forms.

In order to provide overall security, the commerce layer sets up security contexts called 'deals' which provide secure communication and signal certain commerce security attributes to all ongoing protocols. An example for a security service signalled by a context is anonymity: Anonymity can not be provided from individual actions but is a service which needs support by all layers starting with anonymous communication provided by the secure communications block on the supporting services layer.

Since one cannot fix the set of services in advance, the commerce layer includes services for secure downloading of services. This allows customers to participate in business scenarios they never encountered before. Since arbitrary terms of business may be implemented in a new commerce service, a downloaded service need not be secure at all. Security of the implemented services can only be ensured by a separate evaluation, e.g., by trusted consumer organizations who issue certificates on fair commerce services. The secure downloading process together with trust management and access control then ensure that

- each merchant fixes the terms of business in advance, in a non-reputable way,
- that each merchant keeps to its own terms during the whole business, and
- hat services which have not been evaluated by a trusted authority cannot do any harm.

### 3.3. Exchange services

The *Exchange Layer* provides services for fair exchange of business items: Both participants input a business item to be sent as well as a description of the business item expected in exchange. The items are exchanged if and only if both expectations can be met by the item input by the peer. The optimistic protocols [1] proposed by ***SEMPER*** use a third party in case of faults to restore fairness. In the ***SEMPER*** framework, fair exchange should be independent of the actual items exchanged. This is achieved by defining a minimal set of 'exchange-enabling properties' which are required to be implemented by transferable goods in order to guarantee fairness:

- *External Observability*: The third party is enable to check whether a transfer (e.g. sending a message via the third party) was successful or not.
- *Revocability*: The third party is able to undo a transfer (e.g. revoking a credit-card payment).
- *Generatability*: The third party is able to redo a transfer (e.g. signing a replacement receipt).

The exchange manager then negotiates with its peer which of the generic fair exchange protocols shall be used based on the exchange-enabling properties of the two goods to be exchanged. An example for such a protocol is described in Section.

### *3.4. Transfer service*

The *Transfer Layer* provides services for packaging and trading business items. It implements the transfers of the electronic commerce model. The basic items are electronic payments, credentials, and general statements which include digital signatures and data. These business items can be bundled in tree-like packages called *containers*. The security attributes attached to each transfer determine the level of security which is required for the transfer or exchange of the transferred container.

Each type of item is managed by a separate manager which provides unified services integrating existing implementations. The payment manager for example provides three generic services for handling account-based (which includes credit-card payments) and cash-like payments together with the negotiation of the means of payment. Several payment systems of each of these classes can be installed. During a payment, the payer and the payee's payment manager then automatically negotiate which payment system shall be used based on the preferences of the users.

Furthermore, the transfer services define the interfaces of the properties enabling fair exchange. These interfaces can then be supported by any good which may be plugged into a fair exchange protocol. Note that some properties may be trivial for some items while being unachievable for others: For messages, generatability is trivial while revocability is impossible. For signatures, generatability signatures requires additional cryptographic protocols [2].

### *3.5. Supporting services*

The *Supporting Services* provide user preference management, persistent object storage, communication, crypto services, and other supporting services such as access control. Furthermore, this layer provides secure communication services, i.e., a secure connection guaranteeing a given set of security attributes such as anonymity, authenticity, and confidentiality. The design of the anonymous communication service is describe in more detail in Section 4.2.

## 4. Recent updates to the framework

### *4.1. A generic fair exchange protocol*

We now describe an example for the exchange protocols used in *SEMPER*. The protocol can be used to exchange generatable goods (i.e. goods which can be reproduced by the third party) for any observable good (i.e. goods where the third party can verify that the transfer was successful).

The fair exchange protocol (see Fig. 5) is similar to the protocol described in [1]: The basic idea is that the participants first agree on the exchange. If they agree, the responder transfers its good. If the good matches the expectation of the originator, the originator then sends its good as well. If the originator misbehaves and does not send its good, the responder complains at the third party which then produces an equivalent

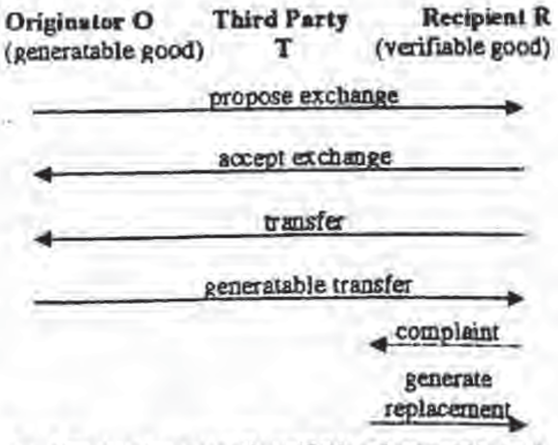


Fig. 5. Exchanging externally verifiable and generatable goods.

replacement for the good (this can be done since the good was generatable). A more detailed description will be published in [8].

After the same pattern, other protocols can be built which guarantee fairness if one of the items provides generatability and the other external observability or if both offer generatability or revocability.

### 4.2. Anonymity and anonymous communication

In *SEMPER*, anonymity services have to be provided by all layers. This is determined by a global anonymity security attribute: If this attribute is set, the supporting services layer provides anonymous communication by default. Furthermore, all services of higher layers only communicate via this anonymous connection and select anonymous protocols during negotiations.

The design of the *SEMPER* anonymous communication module uses a concept of layered secure communication sessions with multiple session oriented MIXs [3] to provide client anonymity[1]: In principle, the client opens a secure connection to the first MIX. Then, the first MIX opens a secure connection to the second MIX and redirects incoming messages. Then, the client opens a secure session with the second MIX. The recipient then acts like the last MIX in the chain. This ensures that the first MIX, for example, cannot read messages sent to the second MIX and thus, is not able to determine the recipient of a MIXed connection (Fig. 6).

## 5. The *SEMPER* trials

The purpose of the trials is to evaluate the applicability and the soundness of the security architecture and services proposed by *SEMPER*. A map of the trial sites is depicted in Fig. 7. The trials are based on the *SEMPER* software prototype which implements the architecture and the basic security services. The trials are being conducted in various business contexts, both internally and externally. Trial evaluation includes interviews of the trial users to measure the degree to which software meets perceived security requirements, to gain information about users' levels of understanding and trust in security options and in order to obtain input to the advanced trials. Only the trials related to the basic services of *SEMPER* are discussed here. An advanced prototype will be evaluated at the end of the project.

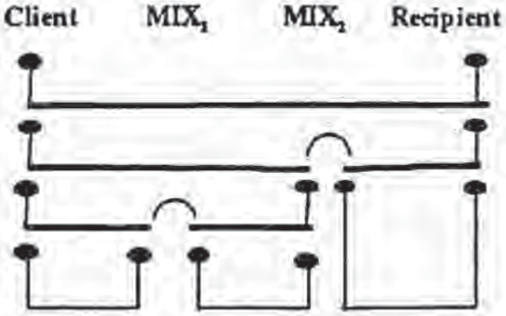


Fig. 6. Anonymous communication (thick lines represent secure and thin lines insecure communication, half circles denote message forwarding).

Fig. 7. SEMPER trial sites.

### 5.1. Internal *SEMPER* trials

In July 1997 the *SEMPER* prototype was tested by two service providers which are members of the project, Eurocom and Fogra. The Eurocom site, located in Athens (Greece), offers distance learning services. Eurocom intends to use *SEMPER* to enable students to browse their offering of courses, register and pay on-line and, subsequently, gain online access to the selected course presentation, notes, and examinations. Fogra, a research institute for the printing industry, located in Munich (Germany), offers its customers on-line ordering and delivery of

---

[1] In the current version, we assume that sellers are not anonymous.

documents and software. In the advanced trials, Fogra plans to offer on-line consultancy.

The 'Basic Trial' *SEMPER* software was used for the supervised trials which took place at Eurocom's premises. The trials were integrated in a series of seminars for SME employees with the title '*Conducting Business over the Internet*'. After a brief presentation of the *SEMPER* architecture, the participants were able to run the *SEMPER* software and to make a purchase of a seminar from the Eurocom electronic store. Due to constraints of time, the *SEMPER* client software was pre-installed and user registration had also been performed prior to the trials. As a result, the participants experienced only the purse creation and purchasing procedures. Thirty people participated in the Eurocom trials. None of the participating companies currently has a website and less than 1/3 of them advertise on another website, however, 1/3 plan to operate a website in the future. Their current interest in the Internet stems from the fact that they feel it would help them to reach a larger market and to remain competitive. More than half also reported that they would expect electronic commerce to enable them to do business more quickly.

In general, the Greek trial participants did not think that Internet is mature enough yet to perform business-to-business, or business-to-customer transactions, not due to the technology itself, but due to its limited use in their customer base. The lack of legal framework regarding the validity of electronic authentication was also viewed as an obstacle to performing important business transactions over the Internet. However, the general feeling was that these obstacles will be resolved and that electronic commerce has much to offer small and medium-sized enterprises. The *SEMPER* software met most of the participant's current business requirements for traditional commerce, apart from cheques as a form of payment. An electronic cheque is, in fact, planned for implementation in the advanced prototype.

Fogra demonstrated the *SEMPER* trial for three days in June 1997 at the IMPRINTA fair. It was demonstrated using a PC with Win95 and ISDN-access to the Internet through CompuServe. Due to both software and access problems, trial of the scenario was only partially successful. Before releasing the *SEMPER* code for external testing, Fogra remedied the problems which had occurred at IMPRINTA. A separate Fogra trial site was set up as the starting point for their participants and a new version of the trusted user interface (*TINGUIN*), including status bars and other enhancements, was integrated. In July 1997, five persons from the Fogra customer base participated in 'unsupervised trials'.

The functionality and flexibility of the *SEMPER* architecture was greatly appreciated by the Eurocom and Fogra trial participants, but the state of the user interface was considered to be insufficiently developed for the ease of use to which non-specialists are accustomed (e.g. windows applications). As a result a new round of supervised trials, with participants selected on the basis of their networking experience, was conducted.

### *5.2. Supervised basic trial*

In December 1997, the *SEMPER* software was installed at the Institute for Computer Science and Social Studies at the University of Freiburg, Germany and tested by 12 trial participants. Twenty hours of in-depth interview material was recorded and has been analysed as the basis of the trial report, project deliverable D09 – Evaluation of Phase II Trials [7]. In order to obtain a more critical evaluation of the software, trial participants with extensive computing experience (and a minimum of 3 yr Internet use) and a good awareness of security issues were selected. The participants subjected the prototype to particularly thorough testing, checking, for example, the software's response to incorrect input (seed too short, incorrect password entry, attempting to obtain a second certificate from the CA, attempting to continue without inputting the requested information, rejecting offers, etc.) and were favourably impressed by its performance.

The Fogra business application and trial website, mentioned above, was used as the trial site. The trial bank was also run from the Fogra server. The Certification Authority (CA) was provided by the *SEMPER* CA at the GMD in Darmstadt, Germany. Each trial participant initialised the locally installed *SEMPER* software for individual use by entering a personal log-in name and password (these were freely chosen by the participant), completing the registration process and creating one, or more, purses. The registration procedure was based on the participant's personal data (name, organisation, city) which had been submitted to the certification authority prior to the trial in order to simulate off-line personal registration. The participant also entered a personal registration key which the CA had assigned to him/her. It was explained to the participants that, under normal conditions, in order to obtain strong certificates, they would have had to personally visit the certification authority and present proof of their identity (passport or ID card) and, in return, have received their registration key and the fingerprint of the CA for verification during the on-line registration.

Having completed the initialisation process, participants used the *SEMPER* software for their first 'semperised' experience of electronic commerce. They used the prototype to securely identify the Fogra website. They then browsed the Fogra website and selected a digital product (an abstract from the Fogra literature databank). Once they had selected the product they wanted and filled out the order form on the Fogra website, they then requested that their local *SEMPER* software process this order securely. They obtained a digitally signed on-line offer from Fogra. They used their locally installed *SEMPER* software to send a digitally signed order to Fogra and used the purse function of the *SEMPER* prototype to make a (simulated) on-line payment. The abstract was delivered to the participant in the Netscape browser.

As achieving legal certainty in electronic commerce will not only depend on the quality of the security technology, but also on the user's understanding and handling of it, during the initialisation of the *SEMPER* software particular attention was paid to the participants' understanding of the actions they were taking, as well as those factors which influenced their ability to successfully complete the process.

The trusted graphical user interface, *TINGUIN*, where all security relevant communications take place, is the visible and vital link between the user and the *SEMPER* software, as a result, it was subject to particular scrutiny. The credibility of the test for the participants was enhanced by the fact that it was possible for them to check the DOS (Java) window at all times during the test (and many did so). This ensured them that the test was actually *live*, i.e. that they were really exchanging certificates and containers with the CA and the bank and website servers in Munich.

### *5.3. SME trials*

Currently, external trials are being conducted by SMEs in The Netherlands and France. They are also supported by the *SEMPER* basic security services and, in contrast to the previous trials, include real on-line payment. The OPL site (Oilfield Publications Limited – http://www.oilpubs.com/semper) is supported by *SEMPER* partner KPN and associate partner Maris (The Netherlands). OPL is offering books, maps, documents, and database access for the oil and gas industry. On-line payments using credit cards and stored-value smartcards have been implemented. Two additional sites, supported by *SEMPER* partner, IBM France, are located in Sophia Antipolis (France). ACRI (http://eurosud1.eurecom.fr/acri mall) provides access to a databank of satellite images, processed and marked up with simulation results, e.g. the evolution of a polluted area, forest fires, etc. Communication is via ATM. The second site is an electronic shopping mall operated by Cicom. The *SEMPER* trial site is ActimÈdia (http://www.cyber landpro.com), which will be selling CD-ROMs. Payments are to be made by credit card.

## Acknowledgements

This work was supported by the ACTS Project AC026, *SEMPER*. However, it represents the view of the authors. *SEMPER* is part of the Advanced Communication Technologies and Services (ACTS) research program established by the European Commission, Directorate General XIII. This description is based on joint work of the *SEMPER* consortium. It is a pleasure to thank all of them for their co-operation and contributions. Furthermore, we would like to thank Rüdiger Grimm and Jörg Veit for valuable comments. The *SEMPER* home-page is at www.semper.org.

## References

[1] N. Asokan, M. Schunter, M. Waidner, Optimistic protocols for fair exchange, 4th ACM Conf. on Computer and Communications Security, Zürich, April 1997, pp. 6–17.

[2] N. Asokan, V. Shoup, M. Waidner, Optimistic fair exchange of digital signatures, IBM Research report RZ 2973, IBM Zurich Research Laboratory, Zürich, November 1997

[3] A. Pfitzmann, B. Pfitzmann, M. Waidner, ISDN-MIXes – Untraceable communication with very small bandwidth overhead, 7th IFIP Int. Conf. on Information Security (IFIP/Sec '91), Elsevier, Amsterdam 1991, pp. 245–258.

[4] M. Schunter, M. Waidner, Architecture and design of a secure electronic marketplace, Joint European Networking Conf. (JENC8), Edinburgh, June 1997, pp. 712.1–712.5.

[5] *SEMPER* Consortium, Basic services: Architecture and design, *SEMPER* Deliverable D03, Århus, October 1996.

[6] *SEMPER* Consortium, Survey findings, trial requirements, and legal framework – Results from first year of project *SEMPER*, *SEMPER* Deliverable D05, Hamburg, December 1996.

[7] *SEMPER* Consortium, Evaluation of phase II trials, *SEMPER* Deliverable D09, Freiburg, 1998, in press.

[8] *SEMPER* Consortium, Architecture, services and protocols, *SEMPER* Deliverable D10, La Gaude, 1998, in press, available at www.semper.org.

**Matthias Schunter** has been a researcher in computer science at the Universities of Hildesheim and Dortmund since 1994. In *SEMPER*, he coordinated the design and implementation of the transfer- and fair exchange layers. His research interests include formal modelling of privacy and the design of protocols providing multi-party security. He has participated in the projects CAFE on offline electronic payments, and in *SEMPER* aiming at an open integrated solution for global electronic commerce. Both projects were funded by the European Union. He received a diploma in computer science from the University of Hildesheim. He is a member of IEEE, ACM, and IACR.

**Michael Waidner** received the Diploma in Computer Science and the Doctorate from the University of Karlsruhe, Germany. Since 1996 he has been the manager of the Network Security research group at the IBM Zurich Research Laboratory. His research interests include cryptography, secure protocols, and all aspects of dependability in distributed systems. He is (co) author of about 60 papers in cryptography and security. He is a member of ACM, GI, IACR and SIAM.

**Dale Whinnett** has been a researcher at the Institute for Computer Science and Social Studies, Albert-Ludwigs University Freiburg, Germany since September 1995. Research focus: user requirements and understanding of security technology. She is currently working in the project *SEMPER* and previously participated in the EU Esprit project CAFE on electronic payments at the Institute for Social Research, University of Frankfurt. She received her diploma in Marketing and German Studies from the University of Lancaster, UK.

PD 19-10-1999
360-365 (6)

XP-002162270

# Trust and Electronic Commerce – More Than a Technical Problem

Kornelia Konrad
*TU Darmstadt, Graduate Program*
*„Technology and Society"*
*konrad@ifs.tu-darmstadt.de*

Gerhard Fuchs
*Center of Technology Assessment*
*in Baden-Württemberg*
*gerhard.fuchs@ta-akademie.de*

Jochen Barthel
*Center of Technology Assessment*
*in Baden-Württemberg*
*jochen.barthel@ta-akademie.de*

## Abstract

*In our paper we argue that the building of trust in electronic commerce depends only partly on technical security and the knowledge of security gaps and ways of closing them. It is not only a technical system which is trusted but rather a socio-technical system, including users, business practices and related institutions. We will take a closer look at the concept of trust and its relation to knowledge, describe the current situation in electronic commerce, and analyse different technical approaches, that aim at providing security, and non-technical possibilities to enhance security and trust through institutions.*

## 1 Introduction

Over the past few years the use of electronic commerce applications has spread at a rapid pace. Still, there are many who think that this process needs to be accelerated, or at least the present willingness to experiment needs to be encouraged even further. Initial euphoria about the newly developed technical and economic possibilities distracted from the discussion of potential obstacles to the continued spread of electronic commerce. Today, obstacles which were at first considered irrelevant are being discussed more intensively, for example the problem of security.

Developers of technology emphasise that they pay utmost attention to the security and reliability of products and systems. It is obvious, however, that concern for security and reliability alone does not guarantee the commercial success of electronic commerce applications. Thus, Müller points out that, in order to avoid expensive mistakes, trust also needs to be developed [Müller96]. Obviously, the opinion that trust is primarily built on technical reliability still exists, but there is also a growing understanding among technicians that commercial success implies a much wider array of factors. Trust may well be a crucial variable. Institutional and cultural factors are significant for the development of trust. The term trust is, in fact, increasingly used by those concerned with information security and electronic commerce. The most popular domain for its usage has been research regarding authentication and the infrastructure for public key technology in a networked environment.

For years social scientists have been dealing with the problem of risk. Beck even developed the notion of a 'risk society' [Beck92]. But very little has been written about trust in technical artefacts or systems. Wagner [Wagner94] says that talking about trust in a risk society will arouse the suspicion that one is being apologetic, although a risk society is actually a trust society and the daily use of a great number of different technologies is usually trustful. Our everyday use of technical artefacts and systems is, in fact, not usually based on a thorough knowledge of functional principles and related risks. With our superficial knowledge we expect technical products to work and that faults will be repaired rapidly or replacements found. We not only trust in technical systems, but also in their social context, i.e. the way we usually do business, the reputation of brand names and/or institutions, the due process of law, etc. It is not so much that we trust a specific technology, such as the telephone, but we rather trust socio-technical systems.

Usually, the user (individual or corporate) cannot adequately assess the security of use. There is not only a lack of knowledge, but it is frequently impossible to acquire adequate knowledge about deficits. Nevertheless, these systems are used. The authors of this paper maintain that trust in a new technological application like electronic commerce only develops, if and when it is adequately socially embedded, i.e. the institutions (routines, norms, organisations etc.) which provide an acknowledged framework for its use, have been developed.

## 2 Trust and Ignorance

Trust is a risky investment.[1] The actor who trusts runs risks. He trusts that his expectations will be fulfilled. Although he has no certainty, whatsoever, he acts as if everything he expects and trusts in will, in

---

[1] There are quite a number of definitions for the term trust. The Oxford English Dictionary mentions three. Each one involves a different understanding of trust. Insofar it seems essential to clarify what we mean by trust.

1060-9857/99 $10.00 © 1999 IEEE

fact, come true [Luhman89: 23]. We assume that in cases where expectations are based on rational considerations or a strong belief that everything the actor expects will actually happen, the actor is, in fact, sure and, therefore, this cannot be defined as trust [Luhman89: 16; Simmel92: 393; Gidden94].

Those who know too little are forced to trust. Equally, an actor who knows too much to be able to make a final judgement, also needs to trust. In order to get a clearer idea of what is behind this differentiation, we will distinguish between various forms of ignorance. We aim to show that ignorance does not necessarily create trust, or mistrust, and that knowledge does not inevitably produce more subjective security.

Based on an analytical distinction used by Beck [96: 300-305] we first of all distinguish between unacknowledged and acknowledged ignorance, arguing that trust is only relevant if actors are aware of their ignorance. Otherwise, they have no reason to feel uncertain and, hence, no need to develop trust. For our argument it is irrelevant whether the subjective feeling of security is justifiable, or not [Beck96: 290].

### 2.1 Unacknowledged Ignorance

Unacknowledged ignorance implies subjective security because from the actors point of view there is no doubt that his or her expectations will be fulfilled. As long as one is unaware of problems, there is no need to trust. In these cases there may be risks, but they do not matter to the actor because he is unaware of them.

### 2.2 Acknowledged Ignorance

In the case of acknowledged ignorance we differentiate between (a) acceptance of uncertainty and (b) rational ignorance. Acceptance of uncertainty (a) is an important characteristic of modern societies. Max Weber interpreted the development of modern societies as a process of constantly increasing domination of nature and a growing systemic ability to control the world. It is supposed that knowledge and control are, in essence, possible which implies that respective knowledge is available within a society.

It may be that this is no longer true of the "risk society" [Beck92]. It seems that more and more certainties disappear without being replaced and faith in the continuous and cumulative improvement of security is declining most obviously in science and technology. Acceptance of uncertainty in this context means that actors are aware of the deficits in their knowledge and admit their 'ignorance'. There is a plethora of scientific results concerning the security of information and communication technology. Experts, however, restrict an assessment of the security of specific applications in two ways. First, they talk about relative security, pointing out that absolute security cannot be guaranteed. For the purposes of combating assaults on the security of a system it is only possible to recommend to establish a more or less viable relationship between protective mechanisms and reasonable expenditure. Secondly, they restrict their assessment with respect to the future validity of their statements, i.e. the pace and direction of the future development of technology cannot be anticipated and, therefore, they consider their statements to be linked to the present state of affairs and knowledge.

Technical development, the contexts of use, as well as the way applications are used, cannot be assessed precisely.[2] Developers and operators try to take this into account and aim at developing error-tolerant systems. But it is impossible to anticipate all possible constellations. Various social science papers on the risks of technology show that the contingency of technical development is an insoluble problem [DierHof92; DieHoMa96]. It is no longer possible to rely on experts to develop and implement secure systems. When experts make their decisions they are aware of the fact that there are significant uncertainties.

The second kind (b) of acknowledged ignorance is rational ignorance which is an important mechanism for maintaining the ability to act in a complex and differentiated society. Regarding technical systems there is always the problem of maintaining the ability to act, despite the vast amount of information that is available but cannot be processed. We use many technical artefacts and systems although we do not, in fact, know how they work. We do not even bother to find out how they work. As long as technical systems work this is not a problem for the user. As early as 1913, Max Weber wrote that it is impossible to know everything about the functioning of technical systems under contemporary conditions because we all are dependent on using the achievements of complex systems in our daily lives, whether technical artefacts or social institutions [Weber88: 471-472]. Without trust in persons and in technical systems, as well as the related social systems, modern societies could hardly survive [Luhman89: 1; 7-8; Preise95: 270; Wagner94].

There is no uni-directional relationship between knowledge and the creation of trust.[3] Especially the notion that more knowledge will create more trust is misleading. The relationship between knowledge and trust is ambivalent. Knowledge does not automatically create trust. It may also lead to mistrust, as "not only favorable aspects, but also dangers, require familiarity

---

[2] Denning/Metcalfe [97] are very informative concerning the uncertainties of prognoses.

[3] Compare examples discussed by Coleman [90: 175-196]

... to enable a trusting or mistrusting future experience" [Luhman89: 19-20]. Moreover, it must be taken into consideration that trust often means the refusal to collect additional information. In a complex environment, with an immense range of information, the ability to act is secured by trusting and not by being aware of all available information. Trust is an attempt to reduce social complexity.

In our opinion, the recommendation that every individual should know everything about the risks and threats of modern information technology is misleading. It implies that individuals must have at their disposal all the tools necessary to close the gaps in security according to the best existing knowledge, according to requirements and state-of-the-art technology [Dierst97: 59].

Our first conclusion is that it is improbable that trust in security can be achieved by a technical solution alone. Technology without any risks is an illusion. Risk management is what matters; a risk management which is guided by the consideration that we cannot have full knowledge about future developments. If trust is to develop in such a situation more than a seemingly perfect technical solution is necessary.

## 3 Trust, Security and Electronic Commerce

The role of trust can be evaluated further in light of the current discussion surrounding the spread of Electronic Commerce. Surveys show that at the moment there is little trust that electronic transactions are secure. In a survey among German consumers 60 % of those interviewed said they did not trust in the security of electronic transactions[4]. The problem of security is of major importance for commercial users, too. The result of the Electronic Commerce Enquête reported the following items as the greatest hindrances to a further spread of electronic commerce: "a lack of general business methods" (71 %), "regulatory deficits in respect of electronically signed contracts" (70%), "unresolved legal aspects" (66.8 %) and "no secure payment in the WWW" (65.9 %) [MülSch99]. That is, the most important problems are not technical problems, but a lack of institutionalization, on the level of practices of action as well as official regulations. Regarding the obstacles to a more widespread use, other surveys on Electronic Commerce also show that (potential) users and suppliers consider regulatory and security aspects to be of great importance[5]. In addition to an adequate IT-infrastructure, a suitable regulatory framework, an adequate supply of services, trust and acceptance by customers are mentioned as playing a crucial role. A comprehensive regulatory framework is necessary, but not necessarily sufficient to foster trust among consumers. Yet, without this trust, which is difficult to measure in objective terms, Electronic Commerce will be slow to gain wide acceptance. Trust is essential to any commercial transaction. Typically, it is generated through relationships between transacting parties, familiarity with procedures, or redress mechanisms.

In this context, it is interesting to note that higher security standards and expectations are demanded for electronic transactions than for traditional business practices. As a matter-of-fact, the security of a special technology is less significant than the trust in a new kind of business practice, i.e. trust in the functioning of a socio-technical system which replaces familiar and, in part, very insecure business methods. Referring to the hindrances mentioned above, an increase in security is obviously not the only solution to the problem. It is impossible for the individual user to assess all the technical details properly and he or she is no longer willing to accept simple affirmations such as "everything is secure". This is especially true for electronic transactions as, to a great extent, they are no longer embedded in traditional contexts. There is no physical counterpart in the transaction. There may not even be real money and the transaction, as such, is being conducted within a computer network. The normal user, therefore, has to trust in the statements of experts and the functioning of institutions which confirm the trustworthiness of a socio-technical system.

With any new technical application, there must first be a learning process. If a person has little or no experience with a technology (or a new business practice), he or she has no direct "empirical" evidence, which could justify trust or distrust, neither concerning the direct effects as the success of transactions or indirect side effects as the unauthorized collection of personal data. It is evident that with the increasing complexity of a technology the possibilities for direct experience are reduced and indirect experience, usually conveyed by the media, will increase in importance. This creates problems which extend far beyond the technical aspects to the role of institutions acting as mediators of experience. The less experience there is the greater the influence of social communication on the trust becomes, which individuals as well as whole societies have in a technology. The experts we interviewed in a project dealing with security complained that users only take security measures seriously if the respective problems have been discussed in the press. This illustrates the prominent

---

[4] http://www.nua.ie/surveys/?f=VS&art_id=905354587&rel=true

[5] Cp. U.S. Government Working Group on Electronic Commerce. First Annual Report. November 1998. Washington, D.C. GPO 1998

significance of this information channel and its role in communicating information about risk.[6] Of course, social communication as well as social practices are culturally dependent and therefore culturally divergent. Differences may occur on different levels, between nations but also between relatively small social groups.

Trust, however, is clearly only one factor leading to greater acceptance. If there are no alternatives, using certain applications may become obligatory (as is often the case in the corporate context). Sometimes people are forced by rational calculation to use a system they actually do not trust in (e.g. aeroplanes for a significant proportion of people). There are also systematic differences between business-to-business and business-to-customer transactions. In the business-to-customer segment the barriers to a wider acceptance are higher and harder to overcome [Heiden98].

## 4 Technology and Institutions – Different Approaches to the Provision of Security and the Inspiring of Trust

It is, of course, necessary to deal with the technical aspects of trust in a technology and – if possible – minimise the risks of a technology. Yet developers have to avoid concentrating on technical problems alone, while endeavouring to achieve as much security as possible. Whether the communication technology will serve a certain purpose in a secure way not only depends on all technical components functioning reliably, but also on the behaviour of the actual user. Technicians often consider this phenomenon to be a disruptive factor. They complain about the "inappropriate behaviour" of the user regarding security problems. This is the case if, for example, users write down PINs and passwords or transmit them insecurely, or select PINs and passwords that can easily be decoded.

Regarding the reliance on technical security measures, a recurrent phenomenon also has to be remembered. If greater technical security actually leads to greater trust in a technical system, than very often the amount of risks people engage in also gets higher. More trust may lead to a less cautious use though, for example transactions in electronic commerce could become of a higher value, therefore the entire risk of the socio-technical system would remain more or less the same as before.

Institutional aspects have a great influence on the process of trust-building. We define institutions as generally applicable regulation patterns in which norms, habits, conventions and values are manifest, and also as governance arrangements, institutional sectors (e.g. systems of research) or formal organisations. Every technology is attached to a variety of such social institutions. It is for this reason that the use, as well as the further development of a technology, cannot be organized arbitrarily, but is embedded in a social context which is responsible for reducing abuse and for generating and confirming acceptance.

In the following we first want to show, that there is no simple and unambiguous relation between technical improvements and the enhancement of security or trust. Therefore we take a look at different and even seemingly contradictory technical approaches, that aim at providing security. Then we will turn to non-technical possibilities of enhancing security and trust through institutions.

### 4.1 Eliminating the technical potential for abuse

An important approach to creating security is to eliminate the technical potential for abuse and, thereby, to reduce the need for trust. However, this applies only to a certain perspective, which we refer to as the perspective of those who know. These may be technical experts, who know how the technical system works. Often it is also necessary to know some relevant details about the context in which the technical system is implemented and used. Unfortunately, this is not the perspective of most of the users of such a system, particularly everyday users. As mentioned above, it is not even necessarily the perspective of the experts. Due to the fact that the complexity of systems is continuously increasing, they usually have only limited knowledge of the relevant aspects and they are unable to anticipate all the technical developments of the future, or the way users will behave.

### 4.2 Creating technical potential for bargaining security levels[7]

Research has not only focused on eliminating the potential for technical abuse in order to achieve security, but also on opening up technical opportunities, which offer different user groups the opportunity to negotiate their security requirements. In contrast to the situation described above, which is defined by the assumedly legitimate security interests

---

[6] This is also mirrored in a policy statement by the OECD. Regarding the Internet it says: „Rightly or wrongly, a few well-publicized incidents have cast it as a Wild West of roaming bandits, immorality, little governance, and an unreliable infrastructure. While certainly an exaggeration, this image, if left to persist, is likely to mean that mainstream consumers and businesses will not widely adopt e-commerce." [OECD98]

[7] We refer to approaches, which were chosen in the research program "Multilateral Security in Communication" (MülRan99).

of one actor or group and the illegitimate interests of the attacker, this approach is based on the assumption that all relevant actors have equally justified interests, however contradictory they may be, for example the interest of accountability versus anonymity, and more or less the same power to apply their (own) rules of the game.

Evaluated from a sociological perspective this approach is very promising. Technologies are often not used in the same way developers imagined they would be and they are used in a variety of ways in a variety of different contexts and situations. Nevertheless, in the long run we expect the opening up of technical potential will result in fairly flexible and locally differentiated yet institutionalised usage patterns which determine the legitimate use of a technology under certain circumstances and the meaning of acts and messages. This can be interpreted as social closure.[8]

Of course social closure is not restricted to the form of rules of action. Legislative and regulation authorities are examples of institutions, which take on the form of organisations.

## 4.3 Trust despite distrust

If we assume that distrust can never be dispelled completely, since it is impossible to close all security gaps and, thus, eliminate all possible causes of mistrust and we cannot expect users to have a comprehensive knowledge of what can be considered secure or insecure, then institutions play an important role. Just as there are different technical ways to foster trust in electronic transactions there is also a variety of institutions which can serve different functions, all of which could make trust in electronic commerce more likely.

Insurance, or the guarantees and warranties supplied by manufacturers, dealers, service providers or certification authorities, help to limit risks, since they assure, that in case, expectations are not fulfilled, the loss will be small. As a result, they are able to create trust even under the overall condition of distrust. A certain degree of institutionalised distrust in the form of controls may even strengthen trust in institutions.

Another possibility to reduce the probability that expectations are not fullfilled, is regulation. Legislative or regulatory authorities as well as (professional) associations regulate which actions are legitimate and who is permitted to offer certain services. People are likely to comply to the rules, because they take them as an unquestioned pattern of action, especially if the regulations are already established for some time, or because they calculate in a more rational way the advantages and disadvantages. Of course, regulations are usually underpinned by sanctions and the institutions which execute them.

Whereas sanctions come into play, if the question, whose expectations are justified, is already settled, institutions, which serve the purpose to regulate conflicts, are necessary, if contradictory expectations and claims arise.

## 4.4 Delegation of interests

Another important type of institutions acts in behalf of the interests of certain groups, for example users or consumers, who are not able to look after their interests themselves, at least not permanently. Usually these groups lack the necessary knowledge. The institutions may be self-organised or organised for example by the state, as for instance of the Offices for Privacy Protection in Germany. Their tasks are describing and treating problems, lobbying and giving advice to the represented groups. All institutions, which are supposed to support the building of trust have to be trusted themselves. But this is especially crucial for consulting institutions, because those who recur to them have little or no possibilities to make a complaint, if they are deceived.

Obligatory control institutions or institutions for technical evaluation and certification perform a somewhat similar function. Users or consumers may consider them as exercising control or evaluation in their interest, if they have reason to believe that the criteria which guide the evaluation are in compliance with their interests.

## 4.5 Extrapolation of trust

As we already mentioned above, an important problem in the emerging field of electronic commerce, where a lot of new services and technologies are offered or even new institutions are founded, for instance new certification authorities, is the lack of experience as a basis for trust. A business partner or a new service provider or institution may be trusted by the others though, if it is part of an institution, that has already proofed to be trustworthy for other reasons.

It appears that the issue of fostering trust in electronic transactions may call for policy measures as a means of facilitating the development of institutions to provide information concerning the sellers and the development of certification schemes, which should be simple, widely recognised and easily understood. Whether such schemes will emerge through the intervention of public authorities, independent and collaborative efforts of consumer organisations, or the

---

[8] Closure in this context means the process by which facts or artifacts in a provisional state characterized by controversy are molded in a stable state characterized by consensus (SchWer98:19).

initiatives of major financial institutions such as credit card agencies is of secondary importance at the moment and may differ between countries. Product liability, warranties and the right to annul sales are only a few of the components which need to be tackled.

## 5 Summary

The study of the development of trust in the security of electronic transactions is complex. We have shown that the need for trust to compensate for an unattainable level of knowledge is a characteristic trait of modern societies. This compensation cannot be achieved by technology alone, as this would require complete knowledge about the way it functions in a certain social context. The reactor incident in Tschernobyl shattered public trust in the reliability of statements concerning the security of technology and science. Moreover, trust does not focus on an isolated technical application, but on the social context in which it is embedded. Trust-building can be supported by institutions, but there is no easy way out. The building of trust can be a very lengthy process, the outcome of which is very hard to predict. The emergence of trust intermediaries, brokers, and third party services for electronic transactions is imminent. Trust will remain necessary, although precarious.

## 6 References

[Beck92] U. Beck: „Risk Society", London, Sage Publications, 1992

[Beck96] Beck: „Wissen oder Nicht-Wissen? Zwei Perspektiven 'reflexiver Modernisierung'", in: U. Beck, A. Giddens, S. Lash: Reflexive Modernisierung. Eine Kontroverse, Frankfurt a.M., Suhrkamp, 1996, 289-315

[Colemn90] J. Coleman: „Foundations of Social Theory", Cambridge, MA., Belknap Press, 1990

[DenMet97] P. J. Denning, R. M. Metcalfe: "Beyond Calculation. The Next Fifty Years of Computing", Copernicus, 1997

[DierHof92] M. Dierkes, U. Hoffmann (Ed.): New Technologies at the Outset. Social Forces in the Shaping of Technological Innovations, Frankfurt/New York, Campus /Westview, 1992

[DiHoMa96] M. Dierkes, U. Hoffmann, L. Marz: Visions of Technology. Social and Institutional Factors Shaping the Development of New Technologies, Frankfurt/New York, Campus 1996

[Dierst97] R. Dierstein: „Duale Sicherheit - IT-Sicherheit und ihre Besonderheiten", in: G. Müller, A. Pfitzmann (eds): Mehrseitige Sicherheit in der Kommunikationstechnik. Verfahren, Komponenten, Integration, Bonn, Reading, Ma., Addison-Wesley, 1997, 31-60

[Gidden94] A. Giddens: „The Consequences of Modernity", Cambridge, Polity Press, 1994

[Heiden98] M. Heidenreich, Informationssysteme und ihre soziokulturellen Voraussetzungen, in: H.-J. Braczyk, G. Fuchs (eds.): Informationstechnische Vernetzung, Baden-Baden, Nomos, 1998, 103-117

[Luhman89] N. Luhmann: „Vertrauen. Ein Mechanismus der Reduktion sozialer Komplexität", 3d ed., Stuttgart, Enke, 1989

[Müller96] G. Müller: Secure Communication: Trust in Technology or Trust with Technology. Interdisciplinary Science Review, Vol. 21 (1996) / No. 4, December 1996, p. 336-347

[MülRan99] G. Müller, K. Rannenberg (Eds.): Multilateral Security in Communications. Technology, Infrastructure, Economy, München, Reading, Ma., Addison-Wesley, 1999

[MüllSch99] G. Müller, D. Schoder: Electronic Commerce – Hürden, Entwicklungspotential, Konsequenzen. Ergebnisse aus der Electronic Commerce Enquête", Arbeitsbericht Nr. 137/März 1999 der Akademie für Technikfolgenabschätzung in Baden-Württemberg

[OECD98] OECD: Dismantling the Barriers to Global Electronic Commerce. Electronic Document. http://www.oecd.org/dsti/sti/it/ec/prod/dismantl.htm, 9.2.98

[Preise95] P. Preisendörfer: „Vertrauen als soziologische Kategorie. Möglichkeiten und Grenzen einer entscheidungstheoretischen Fundierung des Vertrauenskonzepts", in: Zeitschrift für Soziologie 4, 1995, 263-272

[SchWer98] S. Schmidt, R. Werle: "Coordinating Technology. Studies in the International Standardization of Telecommunications", Cambridge MA, MIT Press, 1998

[Simmel92] G.Simmel: „Soziologie. Untersuchungen über die Formen der Vergesellschaftung", Frankfurt a.M., Suhrkamp, 1992

[Wagner94] G. Wagner: „Vertrauen in Technik", in: Zeitschrift für Soziologie 2, 1994, 145-157

[Weber88] M. Weber: „Über einige Kategorien der verstehenden Soziologie", in: M. Weber: Gesammelte Aufsätze zur Wissenschaftslehre, ed. by J. Winckelmann, Tübingen, J.C.B. Mohr, 1988, 427-474

XP-002162271

PD: 06-01-1998
51 - 61
(11)

# Trust in Electronic Commerce: Definition and Theoretical Considerations

Anil Kini and Joobin Choobineh
Department Of Business Analysis And Research
Texas A&M University, College Station, TX 77843-4217
E-✉: a-kini@tamu.edu, Jchoobineh@cgsb.tamu.edu

## Abstract

*Successful electronic commerce sites allow businesses to create low cost or more efficient channels for product sales or to create new business opportunities. The success and acceptance of most on-line businesses depend on several factors, both technological and social. In this paper, we examine the role of trust in successful use and adoption of electronic commerce applications. We define trust as a belief in the system characteristics, specifically belief in the competence, dependability and security of the system, under conditions of risk. Based on this definition, we develop a theoretical model that identifies the factors that effect users' trust in electronic commerce. The theoretical model presented in this paper will serve as the basis for future empirical studies that will aim to measure the impact of these factors on the development of trust in electronic commerce.*

## 1. Introduction

The Internet is the largest network of computers in the world, linking computer networks from all over the world into a seemingly limitless resource for information. The Internet was conceived over 30 years ago as a project in the U.S. Department of Defense's Advanced Research Projects Agency (DARPA) which investigated technologies to connect computer networks of various kinds. The goal of this project was to develop communication protocols and standards that would allow different computers to communicate across linked networks, even in case of global nuclear war. Beginning in 1969, the first network of four computers was designed and installed, and was called the ARPANET. By 1972, electronic mail was introduced on ARPANET, which enabled users to communicate with each other on the network. From that introduction, e-mail has established itself as one of the most popular applications on the Internet, and is a harbinger of the kind of interaction and communication activity that is commonly associated with the Internet today [15]. The decentralized structure of the ARPANET made it easy for other networks to connect to it, thereby prompting academic, government, and industrial networks to join in, forming a large network that became known as the Internet. This philosophy of the Internet is still the same, with the Internet being developed on the open architecture networking model. In this approach, the choice of any individual network technology is not dictated by a particular network architecture but can be selected freely by a provider and made to inter-work with other networks through a meta level "internetworking architecture" [15]. As we will argue later, this open architecture is both an advantage as well as a disadvantage for Internet based businesses.

In the early 1980's, ARPANET, now commonly called Internet, continued to grow as more and more institutions began to realize the benefits of sharing information and research over computer networks. The National Science Foundation, recognizing the importance of the Internet in research and development, assumed the area of leadership of networking technology from ARPA and planned an advanced network called NSFNET [11]. NSFNET provided members of the U.S. academic community access to the NSF supercomputers as well as to one another. In 1989, after ARPA decommissioned the ARPANET, NSFNET replaced ARPANET as the backbone of the Internet, along with several regional networks consisting of U.S. government funded local and regional networks. This collection of networks, now commonly called the Internet, continued to grow at a phenomenal rate, with the addition of additional networks and backbones by the Department of Energy (DOE) and NASA, and commercial and international growth, averaging around 10% per month for months at a time [11].

### 1.1. Commerce on the Internet

Since the Internet was initially started as a military and academic project, the use of the Internet for commercial activities was prohibited. The NSF's

1060-3425/98 $10.00 (C) 1998 IEEE

Acceptable Use Policy (AUP) restricted the use of the backbone to traffic within and in support of the academic and research institutions. As the net continued to grow, more and more businesses became interested in using the Internet as a medium for conducting business transactions. This led to the development of commercial ventures on the Internet, and more and more commercial networks joining the Internet. Therefore, more and more support for the backbone came from industrial contributions, and with the growing number of non academic users, the Internet became more open to the business community. In 1992, NSF relaxed its Acceptable Use Policy, enabling the conduction of business transactions over the Internet. However, it was not until the development of the World Wide Web that E-commerce applications began to reach significant proportions.

## 1.2. The World Wide Web (WWW)

The World Wide Web was conceived at CERN, the European Particle Physics Laboratory in Switzerland. In an article in the Communications of the ACM, Timothy Berners-Lee, the original developer of the WWW, described the Web as a collaborative medium which would allow collaborators in remote sites to share their ideas on all aspects of a common project [3]. The Web is a complex network of documents that are linked together using the hypertext/hypermedia concept. The documents on the Web are interconnected by using hypertext links and can incorporate diverse media such as sound, video, and animation. It allows users to view information in a universal format, independent of the platform they use to access the Web. The Web is now the fastest growing component of the Internet, with WWW servers being set up at exponential rates. Most businesses are rushing to establish a presence on the Web, either for marketing and sales promotions, information dissemination, or for conducting on-line transactions.

Electronic commerce on the Internet is developing rapidly primarily because of the phenomenal growth of the WWW. The Web allows users to access information on the net using a visual interface that can include multimedia such as sound, images and motion pictures. Most businesses have recognized the huge potential of the Web as a tool for reaching vast audiences all over the world. The massive numbers of people on the Web combined with the fact that the Web can be accessed from most corners of the world make it a very important medium for conducting business. The increased reach of the Web along with the lower cost of setting up and maintaining a Web page as opposed to conventional marketing and selling strategies have induced most of the corporations to set up Web sites. However, the true potential of the net is not as a marketing tool, but as a true economic transaction medium, where the entire process of commerce is conducted over the net, from the initial marketing, to the final purchase and delivery.

## 1.3. The Significance of Trust in Electronic Commerce

Since the Internet is based on open system architecture, trust is hard to develop and maintain. The Interenet was primarily conceived as a research environment to enable researchers to cooperate and share information over electronic media. As stated by Bhimani [4], the Internet was not designed as a commercial environment. It operated on a single domain of trust, while provisions were made to allow remote users' access to critical information on machines. Security generally relied on the users' mutual respect, as well as knowledge considered appropriate on the network. This was reasonable when the number of users on the net was comparatively small, and were typically were research scholars and academicians. However, with the phenomenal growth of the Internet, and changes in the demographics of the users, the net is now a universal community, with users coming from all walks of life. As more and more security breaches occurred due to malicious or innocent attacks, the public opinion on security and trustworthiness of using the Internet for commerce has been towards an attitude of distrust.

Several factors can contribute to the lack of trust in electronic commerce, specifically issues of security, dependability, and competence. In this paper, we will argue that trust in electronic commerce is based on the user's belief in the security, dependability, and competence of the system that he/she is interacting with, especially under conditions of risk. Risk is an important component of trust, because an individual's decision to trust is primarily important when there is some risk of negative outcomes. Therefore, in electronic commerce, trust is important when financial transactions or important personal information is involved. For example, most users are comfortable in using the net for searching for information, and will trust the information obtained from the Web. This is because the situation is primarily of low or non-existent risk, because the possibility of negative outcomes is low. The user is aware the information obtained might be flawed or incorrect, but is aware that he/she can selectively use the information received from the Web. On the other hand, most users are less trusting about sending personal information or conducting financial information on the Web. In a study on where users' place their trust, USA Today reported that only 5% of people surveyed trusted the Internet to send credit card information. In a more recent study, the eTrust Internet Study conducted by the Boston Consulting group, it was found that more than 70% of the people surveyed were concerned about sending private information over the Internet [24]. This suggests that most users do not trust the Internet as a medium for conducting commerce.

Trust is a topic of considerable interest in electronic commerce and is important in ensuring that the true benefits of electronic commerce can be realized. It is essential in ensuring that optimal performance benefits are obtained from the system as well as the user. Even though the infrastructure is now in place for facilitating electronic commerce, customers' trust must be developed to ensure its utilization. Understanding this development of trust and its components is the first step. In this paper, we develop a theoretical model of trust by looking at past research on trust in various diverse fields such as psychology, social psychology, and human-computer interaction. In the next section, we provide a review of related literature on trust.

## 2. Previous Research In Trust Among Humans

Trust between humans has been extensively studied in the sociology and psychology literature. Rempel et.al. [21] claim that trust is one of the most desired qualities in any close relationship. Lewis and Weigert [17] claim that trust is indispensable in social relationships. Shapiro et al. [25] have studied trust in business relationships and conclude that *significant benefits can accrue from trust in business relationships.* The psychology literature identifies the factors influencing the development of trust in humans, and the characteristics that are used to determine the propensity to trust or distrust.

In order to develop our theoretical model of trust between humans and electronic commerce, we use past research on trust between humans involved in relationships to argue the case for studying trust in man machine relationships. The research on human trust is mostly focused on the study of trust in humans either in a relationship or in society. We extend these concepts of trust in humans to develop the model for trust in the human-electronic commerce relationship. As the objective of this research is to understand how trust is developed and sustained in human-electronic commerce relationship, a good starting point is to examine the theories of trust in human relationships.

### 2.1. Trust: Some Definitions and Conceptions

The concept of trust has been studied in a variety of situations. As a starting point in our attempt to define trust, let us define the fundamental meaning of trust, as stated in the Webster's dictionary.

1. *An assumed reliance on some person or thing. A confident dependence on the character, ability, strength or truth of someone or something.*
2. *A charge or duty imposed in faith or confidence or as a condition of a relationship.*
3. *To place confidence.*

The multidimensionality of trust is apparent from these three different definitions. The first definition implies that the person is reliant on another for something, and is depending on the ability of that person to perform the task. A common statement to illustrate this kind of trust will be "I trust you to do this job for me." This definition of trust implies a work relationship between the two individuals. The second definition of trust is used to describe trust between individuals involved in a committed relationship, like marriage. The third definition of trust is a more simplistic definition, wherein the aspects of dependability and reliability are not considered. The statement "I trust my senator" best illustrates this kind of trust. The trust is based on the confidence in the other person, on his ability to take the correct decisions. However, the implication is that the person has no significant individual stake in the outcome of the task.

These three definitions of trust definitions suggest that trust assumes the existence of some kind of relationship between two parties, and an expectation of one person about the other person's behavior in the relationship. In other words, you trust somebody or something based on your expectation of the other person. Central to this definition of trust is the confidence and dependence on the reliability, integrity, and truth of another party.

In order to arrive at a model of trust for this research, we shall start by examining the previous efforts to study trust, as described in several literature streams-psychology, social sciences, political science and economics. Some of the earliest work on trust was conducted by Duetsh [7], where he defines trust formally as an expectation of events, saying

*An individual may be said to have trust in the occurrence of an event if he expects its occurrence and his expectations lead to behavior which he perceives to*

1060-3425/98 $10.00 (C) 1998 IEEE

*have greater negative consequences if the expectation is not confirmed than positive motivational consequences if it is confirmed.*

He illustrates this definition of trust by giving an example of a couple hiring a baby-sitter. Depending on the couple's perception or expectation of the occurrence of events, they will decide to trust or distrust the baby-sitter. The couple considers harm to the baby as a greater cost than an evening out is considered an advantage, and therefore base their trust on their expectation of the baby-sitter's behavior or ability to take care of the baby. Thus one of the requirements for trust is the presence of negative and positive outcomes. The expected loss from the negative outcome is necessarily greater than the expected gain from the positive outcome, and therefore the decision to trust is a nonrational choice. If the reverse was true, then the choice would be just economic rationality.

Thus, Duetsh focused on the motivational component in defining trust as the expectation of the occurrence of an event, requiring the existence of both positive or desirable outcomes and negative or undesirable outcomes. The inherent concept stressed by Duetsh was vulnerability. The trusting person *perceives that he will be worse of if he trusts and his trust is not fulfilled than if he does not trust.* The choice to trust is based on expectations, with the choice having the possibility of negative outcomes. This early definition incorporates the element of risk in the decision to trust. If there is no risk, then the choice becomes a choice of economic rationality. Thus, as Duetsh defines it, trust is only relevant when there is uncertainty involved in the outcome of future events. This situation is especially evident in electronic commerce, because the user is exposed to risk if he/she chooses to trust and engage in commerce over the Internet that if he/she chooses alternate means of commerce. The user is therefore vulnerable and is in a situation of risk that his information can be stolen or otherwise misused.

From this early starting point, there have been several different streams of research in understanding trust between humans. Each literature stream examines the concept of trust a little differently and has different conceptualizations of trust. These different conceptualizations illustrate the multidimensional nature of trust in humans. In the next section, we will look at the different literature streams in the study of trust.

## 2.2. Three Perspectives In The Study Of Trust

Social scientists and psychologists have used different approaches to describe and study trust. Worchel [26] has classified these research streams into three categories, each having a distinct approach in studying trust. The three categories are the perspectives of *personality theorists, sociologists and economists,* and *social psychologists.* In this research we will call these three categories *Individual Trust, Societal Trust, and Relationship Trust.* We shall look at each of these three different categories and discuss their impacts on studying trust between humans and electronic commerce.

**2.2.1. Individual Trust (The Approach of Personality Theorists).** This category of trust focuses on the individual's personality characteristics that determine the readiness of the individual to trust. These researchers focus on trust as a personality characteristic that is shaped by specific developmental and social contextual factors. Lewicki et al. [16] state that trust at this level is conceptualized as a belief, expectancy or feeling that is deeply rooted in the personality, with its origins in the individuals early psychological development. Rotter [22] in his essay on the study of interpersonal trust stressed the need to consider individual differences in the study of trust. He states that trust is a specific characteristic of the individual rather than a generalized characteristic. Therefore, he claims that individual differences, which are the result of earlier condition differences, are of primary significance for investigations involving the development, maintenance and stability of trust.

In order to study the impact of individual differences in trust, Rotter developed an Interpersonal Trust Scale (ITS) that places individuals along a continuum of low to high trust. This scale measures the propensity of individuals to interact with others and trust the other to an extent of expecting the other to fulfill his promise. He claims that people who come from an environment where everybody fulfills their promises would tend to place confidence in the promises of relative strangers, whereas a person who often has been misled would tend to disbelieve the promises of strangers.

This view of trust being a characteristic of our past experiences is substantiated by several other researchers. Bowlby [5] suggests that adult concerns about trust is developed on the basis of early relationships between the individual and the caregiver. Wrightsman [27] states that people develop a personal philosophy about their interaction with other people. One of the fundamental elements of this personal philosophy is the individual's view on trust, indicating whether the individual considers others fundamentally honest or basically immoral and irresponsible. Rempel et al. [21] state that in an individual, trust is a factor of how they were treated previously.

Based on the research of personality psychologists that trust in an ingrained characteristic, we develop our initial contention that an individual's decision to trust is dependent on the individual's specific personality characteristic, the intrinsic trusting nature. We call this characteristic the individual's *Tendency To Trust* (TTT). Personality theorists have found in empirical studies that individuals with a high TTT are more willing to trust others in novel situations. In this research, we contend that an individual's TTT influences the human-electronic commerce relationship, specifically by influencing an individual's decision to trust the system.

**2. 2. 2. Societal Trust (The Approach Of Sociologists And Economists).** This approach to trust considers the development of trust between individuals and institutions. Lewicki et al. [16] describe this trust as a phenomenon between and within institutions and as the trust that individuals put in these institutions. This is a more general societal view of trust, wherein the individual has to trust an institution, such as an organization, or societal structures such as a judicial system or an educational system. They describe institutional trust as the trust that develops when individuals generalize their personal trust to large organizations made up of individuals with whom they have low familiarity, low interdependence, and low continuity of interaction. Lewis and Weigert [17] also approach trust as a sociological topic as opposed to a psychological trait within the individual. They claim that trust is a social reality that is functionally necessary for the continuance of harmonious social relationships. Examples of this kind of trust include the trust that citizens have in the government, patients and clients in doctors and lawyers, students in teachers and educational institutions, etc. This research stream attempts to understand the conceptualization of trust between individuals and institutions with which they interact in everyday life from a social perspective.

Societal trust is fundamental to this research, and is the basis for developing our definition of trust in electronic commerce. We contend that trust in the Internet is a form of social trust, which effects the way we interact with the Internet in everyday life. Earle and Cvetkovich [8] contend that social trust is largely significant in our modern society where the complexity of the society has necessitated trust because individuals do not completely understand the inner workings of the system. In his work on modernity, Giddens [10] says that modern systems depend on trust, and that trust is involved in a fundamental way with the institutions of modernity. Giddens notes that trust in systems takes the form of faceless commitments, by which faith is sustained in the workings of the system. He states that trust depends neither on a full initiation of the processes nor upon the mastery of the knowledge that these processes yield. This is especially true of the Internet, where most users are not technically knowledgeable about the processes that computer networks use, but are willing to use these systems. Therefore trust in the Internet is basically a social phenomenon that is effected by the information that we receive in the media as well as from everyday interactions. Thus our trust is a belief in the institution that is the Internet and is effected by several factors like security, dependability and competence. In a later section, we will define trust in electronic commerce based on our discussion of social trust.

**2. 2. 3. Relationship Trust (The Approach Of Social Psychologists).** The final stream of research in the study of trust is the approach of social psychologists who approach trust as an expectation of the other party in a relationship. These researchers focus on the factors that create or destroy trust in individuals involved in a personal or work relationship. From the early work of Duetsh [7], who defined trust as an expectation of the occurrence of an event, social psychologists have examined the expectations that individuals have in others with whom they are involved in a relationship. Zand [29] suggested that trust is the willingness of one person to increase his or her vulnerability to the actions of another person whose behavior he or she could not control. The individual who is making the decision to trust is dependent on the actions of others for the outcome of the decision. Zand defined trust as the *conscious regulation of one's dependence an another that will vary with the task, the situation, and the person.* Thus Zand emphasized the vulnerability aspect of trust by stating that the decision of trusting was dependent on the nature of the task. The more important the issue is to the individual, the more unwilling he is to give up control over the outcome. Therefore, the decision to trust is now a personal decision dependent on the individual's expectation of the outcome, factoring in the importance of the issue to the individual.

Butler [6] emphasizes the role of trust in relationships as opposed to social trust by saying *trust in a specific person is more relevant in terms of predicting outcomes than is the global attitude of trust in generalized others.* Rempel and Holmes [21] in their study on trust in close relationships have developed a theoretical model for describing trust. They state that trust is a generalized expectancy related to the subjective probability an individual assigns to the occurrence of some set of future events. Schelnker et al. [23] define trust as *the reliance upon information received from another person about uncertain environmental states and their accompanying outcomes in a risky situation.*

In this research, we treat the interaction of the individual with he Internet as a relationship and propose that trust is especially important when the situation is one of risk and vulnerability. Thus the decision to engage in electronic commerce is based on the risk involved in the situation, as well as the amount of vulnerability that the user feels in the situation. Therefore trust is especially important to develop in electronic commerce because the financial or personal nature of the transactions puts the user in a situation of vulnerability and risk.

## 2.3. Application of Trust Categories to Human-Electronic Commerce Trust

In the previous paragraphs, we have identified three categories of trust that have been studied in the literature. However, for the purpose of this study, it is necessary to circumscribe these definitions and arrive at a formal model of trust between humans and electronic commerce. In order to study trust between humans and the Internet, it is first necessary to consider the individual's personality characteristics. In order for an individual to trust the Internet, his/her trusting behavior is key in establishing the initial relationship. Secondly, it is important to consider the human electronic commerce interaction from the relationship approach and to study trust in the system, especially under conditions of dependence and vulnerability. Also, to study trust, we need to develop a working definition of trust. In this study, we view trust in the Internet as a form of social trust, which is based on the individual's beliefs about the system.

## 2.4. Conclusions from Previous Research

Based on research in social psychology and personality psychology, we have developed our initial contention that trust between humans and electronic commerce should be considered as a relationship. Also, we identified the factors that effect the development of trust. An initial representation of the previous research on trust as applicable to this research is shown in Figure 1 below.

The three factors that influence the development of trust are *personality, environment, and risk.* One significant personality trait is individuals' tendency to trust or distrust based on their experiences. Persons who have been brought up in a secure environment are more likely to develop a readiness to trust. Environmental cues that contribute to perception of obtaining positive or negative outcomes affect the level of trust. These cues include information about the competence of the system, or knowledge of behavior of the system in the past.

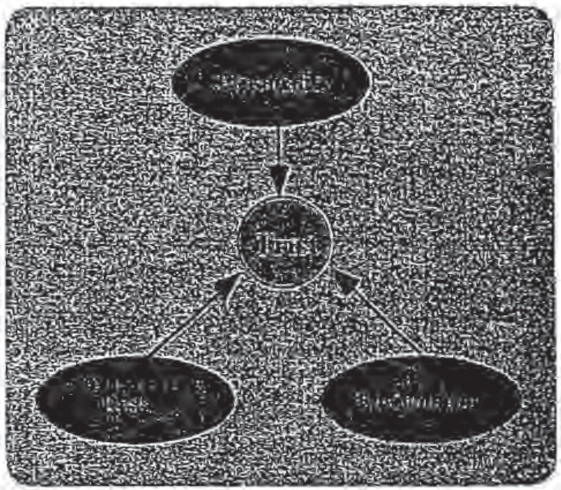


**Figure 1: Factors effecting the development of trust**

One of the most prominent sources of environment cues is the news media like television and newspapers. The Internet has received largely negative publicity about security and dependability in the news media, largely because of the sensational or printworthy quality of break-ins and computer theft. One way of countering the negative perception is by providing positive information about specific aspects of the Internet, such as security issues. The final factor that influences the development of trust is the risk involved in the transaction. The more at stake the person has in the outcome(s), the more difficult it is for him or her to risk trusting the other. Therefore, it is important that for electronic commerce to be widely successful, especially when the risk involved is large, a sufficient amount of trust be developed.

These studies have examined trust between humans or the society. However, in order to completely understand and define trust in electronic commerce, we need to examine research on trust between humans and machines, and then integrate all the studies in a comprehensive model. In the next section, we will look at two research projects that studied trust between humans and machines.

## 3. Models of Trust between Humans and Machines

Muir [18, 19] has developed a framework for studying trust between humans and automated control systems. In this framework, she identifies two dimensions that affect trust between humans and machines. The first dimension in her framework, based on the work of Barber [2], is that of human expectation, specifically the three expectations of *persistence, technical competence, and fiduciary responsibility.* Persistence refers to the belief in the persistence of natural and moral laws and the belief that physical laws are constant, and humans and systems are good and decent. Technical competence refers to the ability of the other partner to provide consistent and desirable performance in their roles. Muir [18] subdivided technical competence into skill based, rule based, and knowledge based behavior, which respectively correspond to expertise in everyday routine performance, technical facility, and expert knowledge. The third expectation, fiduciary responsibility, refers to the expectation that the other partner has moral and social obligations to hold the interests of others above his/her own. According to Muir, these three expectations are the basis for trust between humans and machines. That is, as shown in Figure 2, humans' trust in machines is affected by their expectation of persistence, technical competence, and fiduciary responsibility.

Based on the work of Rempel et al. [21], and in order to represent the dynamic nature of trust, Muir incorporated a second dimension into the model. The dynamism dimension provides a time frame for the development of trust. Trust undergoes predictable changes as a result of experience. This dimension is based on a continuum of expectations, *Predictability, Dependability, and Faith,* which develop systematically as a result of experiences with the system. During the early stages of interaction with a system, a person judges the predictability of the system by assessing the consistency of its recurrent behavior. As experience with the system grows, the trust will be based on the attribution of a dependable disposition, which is based on previous experience. The final stage will be the development of faith in the system. It represents a belief in the dependability of the system's behavior in the future.

In Figure 2, we represent the first dimension of expectation in the inner circle stating that trust in a machine is based on the individual's perception of *persistence, technical competence, and fiduciary responsibility* of the system. The second (dynamic) dimension is represented in the outer circle, with the arrows indicating the sequence of occurrence of the different expectations.

Muir represents these two dimensions as orthogonal counterparts. However, the time based interaction of the two dimensions is not very clear. It is clear that the dynamic nature of trust develops from predictability to dependability to finally faith in the system. However, when crossed with the orthogonal dimension, then the dynamic nature seems to be undermined. Does predictability first affect fiduciary responsibility, or technical competence or persistence? Does dependability affect technical competence before predictability? These are questions that can be raised and remain unanswered in the model. Therefore, Muir's representation of the dynamic or time based dimension as orthogonal to Barber's [2] dimensions of expectation is subject to another interpretation.

Lee and Moray [14], claim that Muir's two dimensions are more complementary than orthogonal. They claim that Rempel et al.'s dimensions of predictability, dependability and faith are really a developmental progression only because of the abstraction required to represent each dimension. They claim that Barber's and Rempel's dimensions correspond closely and propose a different model of trust. They represent trust between humans and automatic control systems with a slightly different framework. They introduce four dimensions: *Foundation, Performance, Process, and Purpose.* The first dimension, Foundation, corresponds exactly to the dimension persistence of Barber and represents the fundamental assumption of natural and social order that makes other levels of trust possible. The second dimension, performance, represents the expectation of consistent, stable, and desirable performance or behavior. The third dimension, process, depends on the understanding of the underlying qualities or characteristics that govern behavior. With machines, this implies understanding the underlying control algorithms, or the knowledge base structure. The final dimension, purpose, represents the underlying motives or intent of the human or the system. These four dimensions represent the dynamic nature of trust and its progression from one stage to the next as experience with the system expands.

Both Muir and Lee and Moray have conducted empirical investigations to evaluate the impact of trust in automatic control systems. They have found that trust is a significant factor in determining operators use of the system. Muir [19] has found that operators subjective ratings of trust and the properties of the automation that determine their trust play a significant part in their use

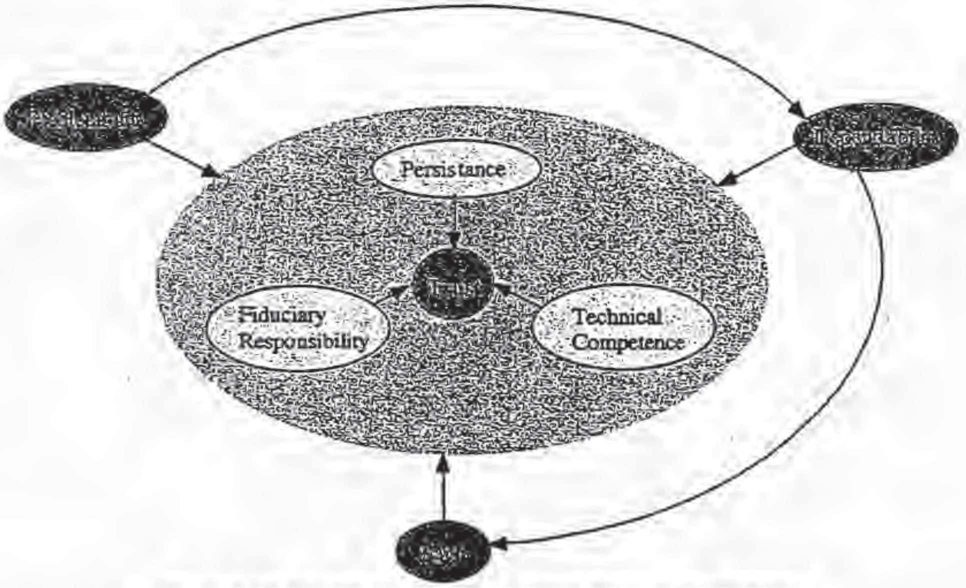


Figure 2: Muir's [20] model of trust between humans and machines

of the automation. Her findings suggest that operators will use the automation only to the extent that they trust it; if they do not trust the automation, they will reject it, preferring to do the task manually. She found that trust in an automation is strongly correlated with the competence of the automation. Thus, the mostsignificant component in her model that affected trust was technical competence.

Lee and Moray [14] conducted a slightly different experiment in which they studied trust as a mediating variable between the properties of the automation and the operator's allocation based on their self confidence. They established that trust is indeed a causal variable, influencing the operators use of the automation. They also found that the individual's self confidence had a significant impact on their allocation of functions to the automatic control system.

These two research works are significant to our present research in two ways. First they establishe the importance of studying trust as a variable in determining the success or utilization of Internet based electronic commerce. Based on their empirical findings, we defend our claim that trust is important in determining individual's decision to engage in electronic commerce. Also their studies have shown that individual factors affect the interaction between humans and automation. As suggested by Muir [19], since individual's trust is based on personal characteristics and knowledge, information provision can result in enhanced trust in the system. Also, their results have found that trust is primarily affected by the technical competence and dependability of the automation. Therefore it provides us with a starting point for developing our definition of trust between humans and electronic commerce. In the next section, we will define trust in electronic commerce based on our previous discussion on research in trust.

## 4. A Definition of Trust In Electronic Commerce.

In order to arrive at a definition of trust in systems, it is first necessary to define the perspective in which trust is being considered. As discussed in the literature review, trust has been conceptualized and defined in several different ways by different researchers based on their particular disciplines. In this research, we borrow from social psychology research on trust in order to arrive at a definition of trust. The justification for this choice is based on the fact that in this study we are interested in studying trust in a social phenomenon, the Internet or any computer based system. Therefore, it is important to consider the view of social psychologists and integrate this with the specific studies on trust in machines to develop a definition that can be empirically validated.

In this research, we adopt the position that trust in a system is a belief that is influenced by the individual's opinion about certain critical system features. Trust in turn, as a belief effects our attitude towards the Internet and influences our decisions to engage in electronic commerce. In order to present this discussion it is important to define and characterize the terms used. Since trust is defined as a belief that effects our attitude, it is important to understand what attitude is and is not. Broadly defined, an attitude is a mental and neural state of readiness, organized through experience, exerting a directive or dynamic influence upon the individuals response to all objects and situations in which it is related [1]. Kerlinger [12] defines attitudes as enduring and organized structures of social beliefs that predispose individuals to think, feel, perceive, and behave selectively toward referents of attitudes. He states that an attitude in effect is a predisposition to behave, and is an unobservable construct, or a latent variable. An attitude is focused on objects in the environment, in this case on the Internet in general, and Electronic Commerce in particular.

An attitude as defined previously, is based on beliefs A belief is defined as statements or propositions that express presumed knowledge, faith or opinion [12]. The three kinds of beliefs, knowledge beliefs, faith beliefs and opinion beliefs form the cores of attitudes. In order to measure attitude, we need to ask people questions on their beliefs. In this research, we are more concerned with knowledge beliefs and opinion beliefs, as they are the ones which can be empirically tested. However, it is important to note that the classification of beliefs is not used in this research, and when we use the term beliefs, we refer to belief as defined in its entirety by [12].

Based on this definition, we explore the interrelationships between attitudes and beliefs a little further by considering attitude theory and behavior theory. As defined previously, beliefs effect our attitudes. Thus beliefs can be viewed as a measure of the probability dimension of a concept. Fishbein [9] states that valid and reliable measures of belief can be obtained by having the subjects judge the concept on a series of bipolar probabilistic scales (e.g., likely-unlikely). However, this definition, while stating that beliefs can be measured, appears to suggest that beliefs are only concerned with the probability of existence of an object. In this research, we are more concerned with the belief of the individual about the object, the Internet. Fishebein clarifies this concept further by distinguishing between two types of beliefs:

a. Belief in an Object: or more completely belief in the existence of an object. This is as defined previously and is an indication of an individuals opinion about the probability of existence of the object.

b. Belief about an Object: This is a belief in the existence of a relationship between the object and some other object or quality. Belief about an object deals with the nature of the object and the manner in which it exists. In general, a belief about an object is the probability or improbability that a particular relationship exists between the object and some other concept, value or goal.

In this research we are primarily concerned with belief about an object, the Internet and its relationship with certain dimensions such as security. Thus, in defining trust as a belief, we are referring to trust as a belief about the Internet and trying to examine the components of trust that contribute to the belief. The definition of trust as used for this research is based on our previous discussion of trust in previous research and is given below:

***Trust in a system is defined as an individuals belief in the competence, dependability, and security of the system under conditions of risk.***

The three components of trust, competence, dependability, and security, effect an individual's trust in the system and therefore his/her decision to interact with the system. This definition is the basis of our research on trust in electronic commerce. It is used to measure an individual's trust in the system with which he/she is interacting. As stated previously, this definition allows us to measure an individual's trust in the system as a composite set of beliefs in the characteristics of the system. In the next section, we present our integrated model of trust in a system, identifying the factors that effect trust in the system.

## 5. An Integrated Model of Trust

In order to study trust in electronic commerce relationships, a theoretical model is developed in this research. The objective in developing this model is to provide a strong theoretical foundation for evaluating the factors that influence trust in electronic commerce.

Our model of trust consists of four dimensions as shown in Figure 3. Fundamentally, trust in an on-line system is a function of the characteristics of the person making the transaction, the on-line system, the task for which the system is being used, and the information environment. All these dimensions influence the development of trust in the system. We will discuss the rationale behind each of these dimensions briefly.

Previous research has shown that an individual's personality characteristics determine the readiness of the individual to trust. Researchers who have studied

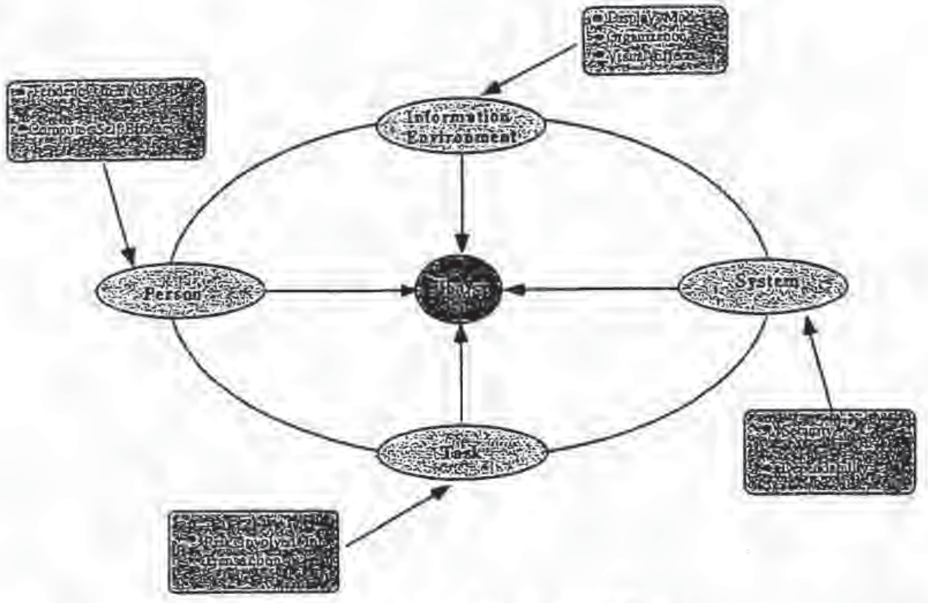


Figure 3: An Integrated Model Of Trust

individuals' trusting behavior contend that the readiness to trust is shaped by specific developmental and social contextual factors. We call this characteristic Tendency to Trust (TTT). Research has shown that when confronted with novel situations, people with a high TTT are more willing to trust others. Also, several researchers have demonstrated that an individual's computer self efficacy determines usage of the system, as it effects the ease of use perceptions. In a recent survey, Yan et al. [28] developed risk factors for several different services offered by banks. Their study shows that most people consider cash transactions on the Internet as the most risky. It is important to study the kinds to task that necessitate trust and to focus on means of fostering and developing trust in these tasks in order to ensure that electronic commerce systems can be developed for a wide range of applications.

The characteristics of the system that the user interacts with are critical in developing and maintaining trust. Several studies have shown that security is a main factor in the success or failure of on-line businesses. Other important factors in the development of trust are the user's perception of dependability and reliability of the system. The user's perception and belief of these three components are the basis of our definition of trust in the system and are discussed in section 4.

The information environment can be seen as two different entities, the environment presented by the system, and the external environment like news media. The environment presented by the system should be correctly perceived and understood by the user. Several studies have shown that presentation and organization of the information are critical in successful adoption of technology. The effects of the information presentation environment on the development of trust should be studied to guide implementation and interface design issues. Specifically, it is important to identify if different presentation modes such as frames and multimedia have an effect on trust in on-line systems. The external environment influences trust by providing knowledge or information about various aspects of the Internet, and contributes to our overall beliefs about the trustworthiness of the system. It is important to understand if trust in a system can be manipulated by providing information about relevant aspects of the system.

## 6. Conclusions and Present Work

In this paper, we presented an integrated model of trust in electronic commerce. This model serves as the theoretical foundation to study the impact of trust on the success of electronic commerce. The model was developed by using past research in diverse fields such as psychology, social psychology, relationship theory, and human machine interaction. The factors that

BEST AVAILABLE COPY

influence trust in electronic commerce were identified and their influence on trust was discussed. The four factors that affect trust in electronic commerce are the individual, the system, the task, and the information environment.

A definition of trust in electronic commerce was also presented. This definition serves as the basis for measuring trust. In order to empirically validate an individual's trust in the system, it is necessary to have an instrument to measure trust. We have developed an instrument based on this definition of trust. To validate the instrument, it was administered to approximately 200 subjects. Exploratory factor analysis was conducted to identify the components of trust. The factor analysis results established the three components of trust, namely: security, dependability, and competence. The results of this analysis are forthcoming in a related paper.

The instrument used in this research was fine tuned for a hypothetical Internet banking application. Several avenues for future research are foreseen here. First, our model of trust has the potential of being extended or modified by yet unforeseen variables for specific electronic commerce applications. Second, the instrument which was developed for this research can be modified to measure trust in these other applications. Third, empirical evaluation of the impact of the factors need to be conducted to measure the influence of the identified factors for each specific application. This will help establish a comprehensive list of factors that positively or negatively effect trust, which can be used to guide implementation strategy for new electronic commerce applications.

## 7. References:

[1] Allport, G. "The Historical Background Of Modern Social Psychology" In G. Lindsey (Ed.). *Handbook of Social Psychology*, Vol. 1, Reading, Mass: Addison Wesley, 1954.

[2] Barber, B., "The Logic And Limits Of Trust" *Rutgers University Press*, New Brunswick, NJ, 1983.

[3] Berners-Lee, T., Cailliau, R., et al., "The World-Wide Web" *Communications of The ACM*, Vol. 37, No. 8, 1994.

[4] Bhimani, A., "Securing the Commercial Internet " *Communications of The ACM*, Vol. 39, No. 6, 1996.

[5] Bowlby, J., "Attachment and Loss, Vol. 2. Separation: Anxiety and Anger" *Hogarth Press*, London, England, 1973.

[6] Butler, J.K., "Toward Understanding and Measuring Conditions of Trust: Evolution of a Conditions of Trust Inventory" *Journal Of Management*, Vol. 17, pp. 643-663, 1991.

[7] Deutsch, M, "Trust and Suspicion" *Journal of Conflict Resolution*, Vol. 2, No. 4, pp. 265-279, 1958.

[8] Earle, T.C., and Cvetkovich, G.T., "Social Trust: Towards A Cosmopolitan Society" *Praeger Publishers*, Westport, CT, 1995.

[9] Fishbein, M., "A Consideration of Beliefs and Their Role In Attitude Measurement" IN Fishbein M. (Ed.). *Readings In Attitude Theory and Measurement*, John Wiley and Sons, NY, 1967

[10] Giddens, A., "The Consequences of Modernity" *Stanford University Press*, Stanford, CA., 1990.

[11] Kahn, R.E., "The Role Of The Government In The Evolution Of The Internet" *Communications Of The ACM*, Vol. 37, No. 8, 1994.

[12] Kerlinger, F.N., "Liberalism and Conservatism: The Nature And Structure Of Social Attitudes" *Lawrence Erlbaum Associates*, Hillsdale, NJ, 1984.

[13] Lee, J., and Moray, N., "Trust, Control Strategies, And Allocation Of Functions In Human-Machine Systems" *Ergonomics*, Vol. 35, pp. 1243-1270, 1992

[14] Lee, J., and Moray, N., "Trust, Self-Confidence, And Operators' Adaptation To Automation" *International Journal of Human-Computer Studies*, Vol. 40, pp. 153-184, 1994.

[15] Leiner, B.M., Cerf, V.G., et al., "The Past And Future History Of The Internet" *Communications of The ACM*, Vol. 40, No. 2, pp. 102-108, 1997.

[16] Lewicki, R.J., and Bunker, B.B., "Developing and Maintaining Trust in Work Relationships" In Kramer, R.M., and Tyler, T.R. (Eds.), *Trust In Organizations: Frontiers of Theory and Research*, Sage Publications, CA, 1995.

[17] Lewis D.J., and Weigert, A., "Trust As A Social Reality" *Social Forces*, Vol. 63, No. 4, pp. 967-985, 1985.

[18] Muir, B.M., "Trust Between Humans And Machines, And The Design Of Decision Aids" *Int. Journal of Man-Machine Studies*, Vol. 27, 527-539, 1988

[19] Muir, B.M., "Trust In Automation Part II: Experimental Studies Of Trust And Human Intervention In A Process Control Simulation" *Ergonomics*, Vol. 39, No. 3, pp. 429-469, 1996.

[20] Rempel, J., and Holmes, T., "How Do I Trust Thee" *Psychology Today*, pp. 28-34, February, 1986.

[21] Rempel, J.K., Holmes, J.G., and Zanna, M.P., "Trust in close relationships" *Journal of Personality and Social Psychology*, Vol. 49, pp. 95-112, 1985.

[22] Rotter, J.B., "Generalized Expectancies For Interpersonal Trust" *American Psychologist*, pp. 443- 452, 1971.

[23] Schlenker, B.R., Helm, B., and Tedeschi, J.T., "The Effects of Personality and Situational Variables on Behavioral Trust" *Journal of Personality and Social Psychology*, Vol. 25, No. 3, pp. 419-427, 1973.

[24] Scott, S., et.al. "Survey Reveals Consumer Fear Of Privacy Infringement Inhibits Growth of Electronic Commerce", *eTrust Project*, 1997. (http://www.etrust.com/press/article003.html).

[25] Shapiro, D.L., Sheppard, B.H., and Cheraskin, L., "Business On a Handshake" *Negotiation Journal*, Vol. 8, No. 4, pp. 365-377, 1992

[26] Worchel, P., "Trust and Distrust" In W.G. Austin and S. Worchel (Eds.), *The Social Psychology of Intergroup Relations*. Belmont, CA: Wadsworth, 1979.

[27] Wrightsman, L.S., "Social Psychology in the Seventies" *Brooks/Cole*, Pacific Grove, CA, 1972.

[28] Yan, G., Paradi, J.C., and Bhargava, S., "Banking on the Internet and its Applications" Proceedings of the 30th Annual Hawaii International Conference on Information Systems, Vol. 4, Maui, 1997.

[29] Zand, D., "Trust and Managerial Problem Solving" *Administrative Science Quarterly*, Vol. 17, pp. 229-239, 1972.

XP-002162272

P.D. 00-09-1997
p. 118-124 = 7

SUPPORTING ARTICLE

# Trust and Traceability in Electronic Commerce

*Dennis D. Steinauer*
*Shukri A. Wakid*
*Stanley Rasberry*
INFORMATION TECHNOLOGY LABORATORY,
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY,
GAITHERSBURG, MD

■ Electronic commerce (EC) will modify some of the traditional models for the conduct of business. However, it is important that many of the long-standing elements of commerce be replicated in the electronic world. Commerce, electronic or otherwise, requires several elements: trading partners, goods and services, units of exchange (money), transaction infrastructures, and delivery and distribution mechanisms. These elements have been developed over centuries of legal, governmental, technological, and commercial practices and have resulted in a business infrastructure that people understand and trust. We explore two important elements of that infrastructure, trust and traceability, in the context of the evolving EC infrastructure. We look at a number of *trust enhancers*, i.e., technology or other processes that can help increase the level of confidence that people have in electronic commerce. We also examine the concept of *traceability*, an important trust enhancer, in detail. Finally, we discuss some specific technologies that can increase the overall level of trust in electronic commerce.

Commercial practice and common law, developed over several thousand years, provide the context into which electronic commerce must fit, if it is to succeed. Early commerce, one step above the simplest trading of two articles between two people, was conducted face-to-face, possibly in front of a witness for the more complex transactions. The advent of reliable mail service in the eighteenth century, the telegraph in the nineteenth century, and the spread of telephones in the twentieth allowed commerce to be conducted on a remote basis. Computer-based commerce via networks such as the Internet is simply one more step in that evolution.

Prior to the era of remote transactions, money was basically precious metal. The Pound Sterling was exactly that. Because transporting large amounts of precious metal in the service of remote trading was both labor intensive and hazardous, improvements to the banking system were needed to permit keeping of accounts and issuing letters of credit, drafts, checks, and vouchers of various kinds. Money, over the last century, has become disconnected from any underlying metal, and is essentially based on trust in the stability of the issuing country. In recent times financial accounts have been kept almost exclusively on computer-based systems.

Despite pressures for rapid development of electronic methods of conducting traditional business activities, the underlying structures, relationships, conventions, and methods of traditional methods will remain the dominant way of doing business. Electronic methods must be developed to coexist with these traditional methods. When elements of commerce "go electronic," a number of significant changes take place that require trust mechanisms, such as technical methods for protecting the confidentiality and integrity of data.

Over the centuries, the marketplace has developed many mechanisms, conventions, and processes designed to engender and maintain a necessary degree of trust among trading partners and other marketplace participants. As the marketplace grew in terms of the number of participants, intermediaries, size and nature of transactions, and other elements, more *trust enhancers* were needed to maintain user confidence and willingness to participate. These trust enhancers have enabled the development of a large, complex, yet relatively efficient, system of commerce, both domestically and internationally.

### Physical vs. Digital Components

In the electronic commerce environment, any or all of several important physical marketplace components may be replaced with digital or electronic representations or substitutes. Examples include the following:

- —Money ⇒ Digital Money
- —Goods ⇒ Digital Objects such as software and information
- —Trading Partners ⇒ Digital Agents, Clients, or Servers
- —Physical Transaction Mechanisms ⇒ Electronic Data Interchange (EDI) Applications and Networks
- —Physical Distribution and Delivery Channels ⇒ Electronic Delivery

#### THREE EXAMPLES OF ELECTRONIC COMMERCE

(1) *Delivering information for a fee.* Sellers of fee-based access to legal and financial databases have provided their services by dedicated telephone subscription, but the new trend is toward the Internet because of its efficiency and economy. The authentication of the information and verification of the provider become absolutely critical. With legal or financial misinformation, fortunes can be lost. With medical misinformation, lives can be lost. A system is needed so that users of the Internet can obtain authentication of the provider of specific information. Such a system may require a central registry or control point for operating an authentication function.

(2) *Delivering goods ("digital objects").* Examples of the retail sale and delivery of goods includes name-brand computer software packages. Services might take the form of a document-editing or preparation service where the documents include electronic text, hardcopy articles, reports, and books. The buyer of the service will usually be able to recognize the work with respect to its origin and utility. In the case of the software package, however, it may be quite impossible for the buyer to determine whether the package is an authentic version of the program, whether it is complete, and whether it is pirated. A central source or methodology for authenticating such transactions may be needed. Further, it may be necessary to modify the uniform commercial code to take into account the management of ownership rights for digital objects.

(3) *Using the Internet to connect buyers and sellers.* Goods or services can be ordered and paid for electronically but delivered in a physical form. Examples include a software package on CD ROM, a movie on DVD, or a printout of a report. Arrangements for air travel can be completed by electronic means. This mode of electronic commerce may provide the greatest growth potential, after most of the public has access to the Internet. It could provide an almost seamless transparency to inventory control, interactive catalog marketing, order transaction, delivery management, installation, and service.

### Enhancing Trust

The digital form of many electronic commerce elements discussed earlier has the potential of becoming more accurate, useful, and cost-effective than the physical form. The ephemeral, intangible, and modifiable nature of digital entities requires substitutes for the traditional trust mechanisms of traditional commerce. In the electronic commerce environment, some of these trust enhancers will exist; some will need to be modified or replaced with mechanisms or techniques that take the new technology into account.

#### REDUCING RISK

A basic concept in commerce is *transfer of risk.* In credit card operations, for example, vendors accept a certain discount on the amount owed in return for transferring the payment risk to the credit card issuer. This same concept applies throughout the entire system of commerce, electronic or otherwise.

Parallel to the concept of risk transfer is the reduction of liability or potential liability. This may involve using formal, published standards (e.g., the Federal Information Processing Standard (FIPS) for the Data Encryption Standard (DES) to protect information through cryptography) or adopting industry best practices. Both have relevance to the use of information technology in electronic commerce.

#### TRUSTWORTHY PROCESSES

Trust is enhanced if a system or process has proved trustworthy in the past. For example, if our colleagues have dealt successfully with a payment processing and clearing system, it is more likely that we will trust it for new applications.

#### TRACEABILITY

Trust is enhanced if the participants know that the elements of a transaction may be traced from origin to completion. Thus, if there is a problem, discrepancy, or other dispute, it will be possible to work back through each step in the process to determine where the problem occurred or who may be held responsible. Receipts, sales slips and tapes, and "carbons" or other copies are examples of documents that enable traceability (see "Traceability" below).

#### INTERMEDIARIES AND TRUSTED THIRD PARTIES

Trust on the side of delivering payment for goods or services, is facilitated by intermediaries, such as banks. For example, banks issue credit cards that facilitate electronic fund transfers and establish the credit of the buyer. The bank serves as an intermediary, with a contractual obligation to pay the vendor.

### ENDORSEMENTS

When a major firm, the U.S. government, or another respected organization, adopts a certain process, product, or technology for its own use, a substantial effect on others may occur, who may be trying to decide what is adequate for their purposes. In the area of information technology (IT) standards, especially security standards, the adoption of voluntary standards, FIPS, or other technical standards is often seen as a measure of endorsement.

Although not as convincing as knowing that a respected party actually uses a certain product or technology, formal endorsements play an important role in enhancing trust. Using FIPS for DES is a type of formal endorsement, in that the government has adopted the DES for its use. FIPS are technical standards and guidelines developed by NIST and approved by the Secretary of Commerce for use government-wide.

### FORMAL TESTING AND CERTIFICATION

An important trust enhancer in electronic commerce is formal testing and certification of components, products, and systems. These processes can provide the purchaser with a degree of assurance of the quality, reliability, or security of the tested components, products, and systems. Also important, a formal testing and certification process provides a target for a developer to meet, and thereby also demonstrates a basis for trust.

### LEGAL UNDERPINNINGS AND REMEDIES

Perhaps the most important element of all commerce, electronic or otherwise, is the underlying system of commercial law and attendant remedies available to aggrieved parties. This is an entire discussion in itself, and is outside the scope of this paper (it is being worked on by experts).

## Traceability

One of the trust enhancers described above is the traceability of transactions, payments, and measurements, which provides the assurance of fairness and, where needed, methods with which to establish legal proof and redress. Several aspects of traceability are discussed below.

### TRADITIONAL MEASUREMENT TRACEABILITY

The traditional concept of traceability relies upon the ability to compare a given measurement directly or indirectly with a standard reference. In this context, open, well-understood tests are a necessary primary reference for either third parties or vendors to use in testing products, and by certifying authorities to use in certifying products. Attributes of assurance, conformance, and performance are only meaningful if they are derived from the same set of tests with the same results and with estimates of uncertainty. Reference tests are key to promoting quality markets and a related testing industry. It is important that such tests be available before product development begins so that vendors can voluntarily incorporate testing into the design.

### TRACEABILITY VS. ACCOUNTABILITY

Traceability, the ability to bind a transaction to originating and accepting entities, does not, in itself, provide accountability. Accountability requires both a legal or policy underpinning and the ability to tie an individual or organizational entity to the transaction. For example, if everyone in a department uses the same email account, accountability to the individual user is not possible. Similarly, if authentication mechanisms are ineffective, then it may be impossible to prove that a given individual actually originated a transaction. Effective identification and authentication methods must be employed if trust and traceability are to be achieved.

### TRACEABILITY VS. MONITORING

To some people, the term "traceability" may have a law enforcement implication suggesting, for example, the ability to monitor or track the activities of an individual. While transaction records and audit trails certainly can provide such a capability, this is different from using traceability to verify the accuracy of a measurement or the authenticity of a set of data. We are concerned here only with the latter.

### TRACEABILITY IN ELECTRONIC COMMERCE

In the electronic commerce setting, traceability is critical to verify transactions.

*Goods and services by traditional methods.* For traditional physical goods and services, perhaps ordered and paid for electronically but delivered through traditional physical means, there should be little, if any, change in existing traceability mechanisms (weights and measures, inspection stamps, e.g., USDA Prime, and certification, e.g., UL, ISO 200, etc.). There may be a need, however, for traceability or verifiability of such claims in electronic catalogs. Another possible need for electronic traceability might be in the acceptance of such goods (i.e., signing for, paying for, or logging the delivery of goods or services with an electronic device).

*Digital objects.* A digital object is an addressable module of data and control (processes) that is likely to be characterized by metadata (data types) to facilitate related search and retrieval. Digital objects can represent a physical object, a process (e.g., a transaction), or a piece of information. The object is characterized by defined attributes (e.g., serial number, content, creation date, or owner) and values for those attributes (e.g., serial_number = "45718", creation_date = "1997/05/11", Owner = "dds", or content = "[some string of bits]"). A digital object is, by its very nature, ephemeral and vulnerable to modification or replication. However, through cryptographic processes, it is possible to verify the origin, authenticity, and integrity of the attributes (i.e., contents) of such objects. This enables the traceability or proof of a direct, trustable path from the object's point of origin to the end user.

Digital objects are expected to generate significant activity in entertainment, education, knowledge/design exchange, and general software distribution. Protection against piracy will become more pressing as EC expands in the global marketplace.

Because piracy is a major potential problem, let us broadly define piracy as the act of taking, copying, "plagiarizing," using, publishing as one's own work, selling, distributing, or incorporating into other works, software, or code without the author(s)' permission or without paying required licensing or usage fees. Typically, this starts with making an unauthorized copy of a work. Therefore, many piracy protection mechanisms focus on preventing the initial copying operations. These mechanisms are discussed below in some detail.

There are several things that can be done, depending on what the author, producer, or distributor is trying to achieve. In effect, these reflect the broad strategies of computer security for providing protection for information.

*Deterrence.* The primary deterrence mechanism is, of course, making something illegal. Most people and organizations do not want to break the law or incur associated penalties. The prohibitions must not be perceived as unreasonable or unfair (e.g., prohibition on using software on both desktop and laptop machines, or making a backup copy) and the penalties must not seem draconian. Corporate and government agencies have long had active programs to help ensure that only properly licensed software is used. Other deterrence methods include the use of serial numbers, required registration, and other installation processes. The detection mechanisms (described below), if they are obvious, are also deterrence mechanisms, since would-be pirates can see that they are "leaving tracks."

*Prevention.* There are several methods of preventing unauthorized copying, installing, or use of software.

(1) *Copy prevention.* To prevent making usable copies of the software, special coding, media, or even intentional errors can be incorporated.

(2) *Installation prevention.* To limit software installation, the input of serial numbers, passwords, or the use of hardware devices (i.e., "dongles" that plug into the serial/parallel port of the machine, smart cards or PCMCIA cards with authentication information/processes, or the use of a "key diskette") can be required. All these can be used either to prevent installation or to prevent use of software except by authorized users. Perhaps the most frequently used mechanism today is the use of a serial number or "key." The key is typically printed on the outside of a diskette or CD-ROM, and must be provided by the user at installation time to decrypt the software on the media and to incorporate a unique code in the installed code. This code is then checked by the software itself when it is started. Some software (especially when provided on writeable media) will write information back on the installation disk to prevent more than a designated number of installations. This, of course, is not possible with read-only media such as CD-ROMs. Serialization, encryption, and the digital signing of source software can also be used to ensure that the user has a valid, accurate, and virus-free copy of the software. Server-executable software with minimal client resident modules is another method that prevents installation or use by unauthorized users. This enables the remote server to control the software as well as authorized access.

(3) *Usage prevention.* Most of the methods mentioned above can also be used to prevent the use of installed software. For example, it may be required that a user-unique disk, CD, PCMCIA card, smart card, or dongle be on the machine whenever the software is run. However, except for very expensive, narrow-market software, this requirement is likely to annoy the user. Therefore, most popular systems focus on preventing unauthorized installation (and subsequent copying of the installed software) through unique serialization or another method, and leave it to the user to keep people off his/her system.

*Detection.* As mentioned above, many software vendors focus on detecting (and proving) unauthorized copying or use rather than trying to prevent it outright. In a local network environment, this can prevent use. For example, it is possible for software to check the network for duplicate serial numbers and prevent operation if duplicates are discovered. The use of serialized software also makes it possible to detect unauthorized distribution or mass-copying of software. In addition to enabling software to "self-destruct," this also enpowers an auditor (or law enforcement official) to prove unauthorized copying.

*Recovery.* The normal recovery mechanisms for software piracy are legal actions. It is also possible for a piece of software that detects that it is being misused (i.e., used without authorization) to erase itself or even other files on the system on which it is running. (This is seldom done, because it is rife with liability and criminal potential).

Other possibilities for recovery depend upon the network, web, and electronic mail capabilities on most of today's machines. It is possible for software, during installation or use, to contact a remote site to check registration or other information, or even to send a message to a remote site in the event of detected unauthorized use.

*Economics.* Although there are many ways to deter, prevent, detect, and even recover from piracy, these methods should be incorporated only after a careful analysis of the costs and benefits. For relatively small, special-purpose software, it may even be best to distribute it for "free" and simply encourage "ethical" users to pay a small fee. This is the basis for the wildly successful "shareware" market.

For highly valuable, wide-distribution software (probably the most vulnerable to piracy), many of the techniques discussed above are used. In general, the less intrusive a method is to the user, the more likely it is to succeed, in terms of both increased sales and fewer attempts to "break" the protection. Fortunately, today's software in this category is quite complex and is not installed by simply copying a few files, which makes the installation-point protection mechanisms the most effective for digital objects. All these protection mechanisms rely on the ability to trace the ownership of software.

*Transactions and transaction contents.* Not only must the software objects be protected, but the transaction details must be inviolable. In fact, perhaps the single most important aspect of traceability of electronic commerce activities will be the extensive and comprehensive use of digital integrity methods to ensure that transaction contents, dates of processing, and identities of trading parties are not changeable. This cannot be addressed simply by writing all such information to write-once media, as is sometimes suggested. Rather, there must be assurances that these items are properly bound to user identities and other events in a system. Secure messaging, digital timestamping, and digital signatures are all important technologies for this purpose (see "Integrity" below).

*System components ("pedigree").* The degree of confidence in a product or system component depends upon the user's belief that the items he/she has received are identical to those created or shipped by the system producer. Since software modules may be changed at several points between vendor and user and while on the user's system, there needs to be a way to confirm that they have not changed. Although a simple comparison of a module with its reference version might be possible on a user's system, it is not feasible to do such code comparisons against the vendor's original. However, it is possible for a vendor to sign files digitally, and for the user to confirm the signatures at any time. As long as the user trusts the vendor, this can confirm the "pedigree" of the file, and thereby provide the user with confidence in the product.

Traceability and trust are two important components of the electronic commerce infrastructure. In some cases, trust is accepted as traceability, as with "pedigree." However, traceability is still necessary if the pedigree is questioned.

## Technical Needs

While this discussion focuses on the technical underpinnings of trust enhancers and traceability, it is important to keep in mind that these needs must be combined with traditional procedural, organizational, and physical controls in order to be effective and credible.

### IDENTIFICATION AND AUTHENTICATION

To achieve a level of trust, a system must be secure against unauthorized use—that is, unauthorized access to or modification of system components or data. The most popular form of authentication, the reusable password, should no longer be considered acceptable, in general, for electronic commerce transactions, and especially not for those conducted over open networks, such as the Internet. As the need has grown for more reliable, less vulnerable substitutes, some promising technologies have become cost-effective and popular.

*Smart cards and tokens.* Smart cards and tokens provide portable, "active," and potentially secure devices that can hold sensitive personal authentication data and can often perform sensitive operations that are independent of the computer into which they are inserted. These are increasingly being used to hold keys and certificates; however, they can be stolen, duplicated, or forgotten at home.

*Biometrics.* Technology is now available that enables the cheap, reliable measurement of a physiological characteristic that can be used to verify the identity of a person claiming to be an authorized subject. The accuracy of biometric recognition is measured by determining the percentage of accepted impostors and the percentage of rejected authorized users. There are numerous, critical applications used in government (e.g., entitlements, law enforcement, and immigration), as well as in industry (e.g., access control, access to website servers, access through firewalls, and banking by Internet). An example of authentication technology is the use of fingerprints. Since a hundred years of history have demonstrated their uniqueness, fingerprint technology is cheap, fast, accurate, and easy to use. The binding of a fingerprint biometric to authentication servers via secure communication mechanisms is a key trust enabler. As a matter of fact, the dual use of smart cards with real-time secure fingerprint recognition is a promising technology for identification and authentication.

*Access control.* After users are identified and authenticated, access control mechanisms are needed to enforce the "rules" in a given system regarding the functions and information to which a user will be granted access. In electronic commerce systems, access control mechanisms are being developed that will control access to a given system by development, maintenance, and support personnel, as well as by regular users.

One of the more promising developments is Role Based Access Control (RBAC) technology. Under this type of mechanism, access to functions or data is based on the role of an individual in a given context, not simply by the user's identity. For example, an individual may be authorized at certain times to perform the role of purchaser, and at other times of an approving official, but not at the same time. Moreover, the actual capabilities of purchaser may change from time to time. Thus, access control decisions are better handled through user roles than through strict user identities, although user identity should be maintained to enable traceability and personal accountability.

*Integrity.* Most digital information is inherently "changeable" since most storage and transmission media are reusable or rewriteable. Of critical importance in any electronic commerce system will be the ability to protect the integrity of information and digital objects as they are passed around in and among systems. Maintaining the integrity of audit trails and other transaction data is also a fundamental objective of traceability. In general, this is achieved not by preventing modification of information, usually an impossibility, but by ensuring that any changes can be detected with a very high degree of confidence. The mechanisms to accomplish this are secure message digest techniques and digital signatures.

*Confidentiality.* Not all information in electronic commerce is private. Indeed, by law, many types of transactions must be made available to various parties, ranging from the government to the public. As a practical matter, there will often be several parties to a transaction who must have access to the information. However, there will often be a requirement for some level of confidentiality. Given the open and uncontrollable nature of virtually all types of networks, it is necessary to protect confidential data through encryption mechanisms.

*Government Interest.* The Federal government, a large user of information technology, has a number of possible interests in trust and traceability in electronic commerce. The Clinton administration has made a major commitment to the use of information technology to conduct "the nation's business" and to deliver government services. The following are possible areas of interest:

—conducting government business, among entities of the government, with commercial entities and with other governments (state and international);

—providing government services and meeting government obligations (in social security, taxation, and other functions);

—providing encouragement and incentives to promote the use of electronic commerce between the government and its trading partners;

—promoting research, standards, interoperability, measurement methods, and forward-looking prototypes;

—providing consumer protection and privacy protection;

—protecting citizens' interests and rights;

—supporting law enforcement (collecting, maintaining, and disseminating law enforcement information);

—protecting critical national infrastructures; and

—protecting national security.

## Conclusion

Trust is essential for commerce. As the shift to electronic commerce is made, trust mechanisms must be developed that allow the buyers, sellers, and intermediaries to have confidence in the system. Several trust-enhancing mechanisms have been discussed here, including traceability and technical mechanisms such as identification and authentication, access control, and protection of the integrity and confidentiality of information. The success of the development of trust mechanisms will depend on an effective partnership between industry and government, with the private sector leading. By working together, government and industry can advance the development of electronic commerce systems that are secure, interoperable, and reliable. **sv**

## Acronyms

| | |
|---|---|
| CD | Compact Disk |
| CD ROM | Compact Disk Read Only Memory |
| DES | Data Encryption Standard |
| DVD | Digital Versatile Disk |
| EC | Electronic Commerce |
| EDI | Electronic Data Interchange |
| FIPS | Federal Information Processing Standards |
| IETF | Internet Engineering Task Force |
| IT | Information Technology |
| NIST | National Institute of Standards and Technology |
| PCMCIA | Personal Computer Memory Card International Association |
| PKI | Public Key Infrastructure |
| SET | Secure Electronic Transaction |
| UL | Underwriter Laboratories |
| USDA | U.S. Department of Agriculture |

## Bibliography

*A Framework for Global Electronic Commerce.* 1997. The White House, Washington, DC. July 1.

BERNSTEIN, T. ET AL. 1996. *Internet Security for Business.* Wiley, New York, NY.

FORD, W. 1994. *Computer Communications Security.* Prentice Hall, New York, NY.

*IEEE Computer.* 1997. Special Issue on Electronic Commerce. *30,* 5 (May).

LYNCH, D. AND L. LUNDQUIST. 1996. *Digital Money.* Wiley, New York, NY.

NEGROPONTE, N. 1995. *Being Digital.* Alfred A. Knopf, New York, NY.

PERRITT, H. H. JR. 1996. *Law and the Information Superhighway.* Wiley, New York, NY.

---

*Disclaimer:* Any trademarks in this paper are intended as examples only. No NIST endorsement is implied.

---

This article was written while the authors were employees of the U.S. government.

# EAST Search History

| Ref # | Hits | Search Query | DBs | Default Operator | Plurals | Time Stamp |
|---|---|---|---|---|---|---|
| L4 | 170 | (try near buy) | US-PGPUB; USPAT | OR | ON | 2007/04/26 19:20 |
| L5 | 50 | L4 and (@ad<"19990804" @prad<"19990804") | US-PGPUB; USPAT | OR | ON | 2007/04/26 19:20 |
| L6 | 171 | baum.xa. | US-PGPUB; USPAT | OR | ON | 2007/04/26 19:20 |
| L7 | 64 | L6 and quality | US-PGPUB; USPAT | OR | ON | 2007/04/26 19:20 |
| L8 | 12 | L7 and (@ad<"19990804" @prad<"19990804") | US-PGPUB; USPAT | OR | ON | 2007/04/26 19:20 |
| L9 | 524 | watermark$ and (second near watermark) | US-PGPUB; USPAT | OR | ON | 2007/04/26 19:20 |
| L10 | 84 | L9 and (@ad<"19990804" @prad<"19990804") | US-PGPUB; USPAT | OR | ON | 2007/04/26 19:20 |
| L11 | 27 | L10 and server | US-PGPUB; USPAT | OR | ON | 2007/04/26 19:20 |
| L12 | 26 | L11 and quality | US-PGPUB; USPAT | OR | ON | 2007/04/26 19:20 |
| L13 | 24 | L12 and (low$5 or degrad$) | US-PGPUB; USPAT | OR | ON | 2007/04/26 19:20 |
| L14 | 20 | L13 and remote | US-PGPUB; USPAT | OR | ON | 2007/04/26 19:20 |
| L15 | 20 | L14 and address$ | US-PGPUB; USPAT | OR | ON | 2007/04/26 19:20 |
| L16 | 20 | L15 and stor$4 | US-PGPUB; USPAT | OR | ON | 2007/04/26 19:20 |
| L17 | 20 | L15 and stor$4 | US-PGPUB; USPAT | OR | ON | 2007/04/26 19:21 |
| L18 | 20 | L17 and domain | US-PGPUB; USPAT | OR | ON | 2007/04/26 19:21 |
| L19 | 18 | L18 and authenticat$ | US-PGPUB; USPAT | OR | ON | 2007/04/26 19:22 |
| L20 | 0 | l19 and (try near buy) | US-PGPUB; USPAT | OR | ON | 2007/04/26 19:22 |
| L21 | 0 | l19 and ((try near buy) or demo) | US-PGPUB; USPAT | OR | ON | 2007/04/26 19:23 |
| L22 | 16 | l19 and temp$5 | US-PGPUB; USPAT | OR | ON | 2007/04/26 19:23 |
| S1 | 69 | watermark$ and ((second near watermark$) and (third near watermark$)) | US-PGPUB; USPAT | OR | ON | 2006/10/03 09:14 |

# EAST Search History

| | | | | | | |
|---|---|---|---|---|---|---|
| S2 | 11 | S1 and (@ad<"19990804" @prad<"19990804") | US-PGPUB; USPAT | OR | ON | 2006/10/03 09:15 |
| S3 | 0 | S2 and server | US-PGPUB; USPAT | OR | ON | 2006/10/03 09:15 |
| S4 | 7 | S2 and quality | US-PGPUB; USPAT | OR | ON | 2006/10/03 09:17 |
| S5 | 0 | S4 and legacy | US-PGPUB; USPAT | OR | ON | 2006/10/03 09:16 |
| S6 | 470 | watermark$ and (second near watermark) | US-PGPUB; USPAT | OR | ON | 2006/10/03 09:17 |
| S7 | 80 | S6 and (@ad<"19990804" @prad<"19990804") | US-PGPUB; USPAT | OR | ON | 2006/10/03 09:18 |
| S8 | 25 | S7 and server | US-PGPUB; USPAT | OR | ON | 2006/10/03 09:18 |
| S9 | 24 | S8 and quality | US-PGPUB; USPAT | OR | ON | 2006/10/03 09:18 |
| S10 | 22 | S9 and (low$5 or degrad$) | US-PGPUB; USPAT | OR | ON | 2006/10/03 09:19 |
| S11 | 0 | S10 and (add?in) | US-PGPUB; USPAT | OR | ON | 2006/10/03 09:19 |
| S12 | 19 | S10 and remote | US-PGPUB; USPAT | OR | ON | 2006/10/03 09:19 |
| S13 | 19 | S12 and address | US-PGPUB; USPAT | OR | ON | 2006/10/03 09:19 |
| S14 | 19 | S12 and address$ | US-PGPUB; USPAT | OR | ON | 2006/10/03 09:20 |
| S15 | 19 | S14 and stor$4 | US-PGPUB; USPAT | OR | ON | 2007/04/26 19:20 |
| S16 | 19 | S15 and domain | US-PGPUB; USPAT | OR | ON | 2006/10/03 09:22 |
| S17 | 3 | S16 and legacy | US-PGPUB; USPAT | OR | ON | 2006/10/03 09:20 |
| S18 | 17 | S16 and authenticat$ | US-PGPUB; USPAT | OR | ON | 2007/04/26 19:21 |
| S19 | 17 | S16 and authentic$ | US-PGPUB; USPAT | OR | ON | 2006/10/03 09:34 |
| S20 | 153 | baum.xa. | US-PGPUB; USPAT | OR | ON | 2006/10/03 09:34 |
| S21 | 61 | S20 and quality | US-PGPUB; USPAT | OR | ON | 2006/10/03 09:35 |
| S22 | 12 | S21 and (@ad<"19990804" @prad<"19990804") | US-PGPUB; USPAT | OR | ON | 2007/04/26 19:20 |

# EAST Search History

| | | | | | | |
|---|---|---|---|---|---|---|
| S23 | 10 | ("5195135" \| "5715316" \| "5805700" \| "5845088" \| "5898779" \| "5953506" \| "6026164" \| "6216228" \| "6449718" \| "6557102").PN. | US-PGPUB; USPAT; USOCR | OR | OFF | 2006/10/03 09:35 |
| S24 | 74 | watermark$ and ((second near watermark$) and (third near watermark$)) | US-PGPUB; USPAT | OR | ON | 2007/01/03 09:29 |
| S25 | 0 | S24 and (try near buy) | US-PGPUB; USPAT | OR | ON | 2007/01/03 09:31 |
| S26 | 162 | (try near buy) | US-PGPUB; USPAT | OR | ON | 2007/01/03 09:31 |
| S27 | 50 | S26 and (@ad<"19990804" @prad<"19990804") | US-PGPUB; USPAT | OR | ON | 2007/04/26 19:20 |
| S28 | 23 | S27 and authori$ | US-PGPUB; USPAT | OR | ON | 2007/01/03 09:33 |
| S29 | 2 | S28 and watermark | US-PGPUB; USPAT | OR | ON | 2007/01/03 09:46 |
| S30 | 710 | colvin.in. | US-PGPUB; USPAT | OR | ON | 2007/01/03 09:46 |
| S31 | 13 | S30 and revak.xa. | US-PGPUB; USPAT | OR | ON | 2007/01/03 09:47 |

UNITED STATES PATENT AND TRADEMARK OFFICE

MN

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 10/049,101 | 07/23/2002 | Scott A. Moskowitz | 80408.0011 | 8028 |

7590 05/09/2007

Scott A. Moskowitz
#2505
16711 Collins Avenue
Miami, FL 33160

| EXAMINER | |
|---|---|
| AVERY, JEREMIAH L | |

| ART UNIT | PAPER NUMBER |
|---|---|
| 2131 | |

| MAIL DATE | DELIVERY MODE |
|---|---|
| 05/09/2007 | PAPER |

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

PTOL-90A (Rev. 04/07)

| Office Action Summary | Application No. | Applicant(s) |
|---|---|---|
| | 10/049,101 | MOSKOWITZ, SCOTT A. |
| | Examiner | Art Unit |
| | Jeremiah Avery | 2131 |

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1)☒ Responsive to communication(s) filed on *03 July 2006*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4)☒ Claim(s) *1-31* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-31* is/are rejected.

7)☒ Claim(s) *12* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *08 February 2002* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

## DETAILED ACTION

1. Claims 1-31 have been examined.

2. Responses to Applicant's Remarks have been given.

### *Information Disclosure Statement*

1. The following references were not considered because they were not provided:

EPO Application No. 96919405.9

Japanese Patent Application No. 2000-542907

### *Claim Objections*

2. Claim 12 is objected to because of the following informalities: grammatical errors. In line 7, "means receive a copy...", the word "to" should be inserted between the words "means" and "receive". Appropriate correction is required.

### *Claim Rejections - 35 USC § 102*

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

Claims 1-31 are rejected under 35 U.S.C. 102(b) as being anticipated by United States Patent No. 5,341,429 to Stringer et al., hereinafter Stringer.

3. Regarding claim 1, Stringer discloses a local content server (LCS) for creating a secure environment for digital content, comprising:

a) a communications port in communication for connecting the system via a network to at least one Secure Electronic Content Distributor (SECD), said SECD capable of storing a plurality of data sets, capable of receiving a request to transfer at least one content data set, and capable of transmitting the at least one content data set in a secured transmission (column 3, lines 25-30, "floppy diskette copy protection", column 4, lines 49-57, column 5, lines 35-40 and 53-60, column 9, lines 53-63, "transaction code is given to a vendor sales representative at a remote location" and column 12, lines 13-59);

b) a rewritable storage medium whereby content received from outside the LCS may be stored and retrieved (column 5, lines 35-40 and column 8, lines 39-44);

c) a domain processor that imposes rules and procedures for content being transferred between the LCS and devices outside the LCS (column 3, lines 55-61, "time-limited and/or function limited use of the data", column 4, lines 6-22, column 5, lines 41-48, column 6, lines 4-11, column 8, lines 39-44 and 63-68, column 9, lines 1-13, column 10, lines 60-68", lets customers work with the software on a 'trial' basis (e.g. up to ten times)" and column 11, lines 1-9, "Upon credit approval, the sales representative gives the customer a special code number(s) that 'unlocks' the software products(s) for unrestricted use");

d) a programmable address module which can be programmed with an identification code uniquely associated with the LCS (column 7, lines 43-57, "a watermark or copyright notice that is inserted into the original material" and column 9, lines 43-52); said domain processor permitting the LCS to receive digital content from outside the LCS provided the LCS first determines that the digital content being delivered to the LCS is authorized for use by the LCS and if the digital content is not authorized for use by the LCS, accepting the digital content at a predetermined quality level, said predetermined quality level having been set for legacy content (column 5, lines 61-64, column 7, lines 22-57, "provides a secure system which limits unauthorized access to the materials" and column 13, lines 10-58, "denatured audio that is of adequate quality for evaluation purposes, but not for regular listening" and "VCA drops the amplitude of the source audio signal by 20 dB for a series of 20 millisecond intervals").

4. Regarding claim 2, Stringer discloses e) an interface to permit the LCS to communicate with one or more Satellite Units (SU) which may be connected to the system through the interface, said SUs capable of receiving and transmitting digital content (column 4, lines 33-57, column 5, lines 35-40 and 53-64, column 6, lines 1-3 and 61-66, column 9, lines 43-63, "transaction code is given to a vendor sales representative at a remote location (61), e.g. over the telephone lines (65)", column 10, lines 53-68 and column 11, lines 1-9);

wherein said domain processor permits the LCS to receive digital content from an SECD that is connected to the LCS's communication port, provided the LCS first determines that digital content being received is authorized for use by the LCS (column 4, lines 33-57, column 7, lines 22-33, "provides a secure system which limits unauthorized access to the materials" and column 9, lines 43-67),

wherein said domain processor permits the LCS to deliver digital content to an SU that may be connected to the LCS's interface, provided the LCS first determines that digital content being received is authorized for use by the SU (column 4, lines 33-57, column

7, lines 22-33, "provides a secure system which limits unauthorized access to the materials" and column 9, lines 43-67).

5. Regarding claim 3, Stringer discloses a local content server system (LCS) for creating a secure environment for digital content, comprising:

a) a communications port in communication for connecting the system via a network to at least one Secure Electronic Content Distributor (SECD), said SECD capable of storing a plurality of data sets, capable of receiving a request to transfer at least one content data set, and capable of transmitting the at least one content data set in a secured transmission (column 3, lines 25-30, "floppy diskette copy protection", column 4, lines 49-57, column 5, lines 35-40 and 53-60, column 9, lines 53-63, "transaction code is given to a vendor sales representative at a remote location" and column 12, lines 3-59);

b) an interface to permit the LCS to communicate with one or more Satellite Units (SU) which may be connected to the system through the interface, said SUs capable of receiving and transmitting digital content (column 4, lines 33-57, column 5, lines 35-40 and 53-64, column 6, lines 1-3 and 61-66, column 9, lines 43-63, "transaction code is given to a vendor sales representative at a remote location (61), e.g. over the telephone lines (65)", column 10, lines 53-68 and column 11, lines 1-9);

c) a rewritable storage medium whereby content received from an SECD and from an SU may be stored and retrieved (column 5, lines 35-40 and column 8, lines 39-44);

d) a domain processor that imposes rules and procedures for content being transferred between the LCS and the SECD and between the LCS and the SU (column 3, lines 55-

61, "time-limited and/or function limited use of the data", column 4, lines 6-22, column 5, lines 41-48, column 6, lines 4-11, column 8, lines 39-44 and 63-68, column 9, lines 1-13, column 10, lines 60-68", lets customers work with the software on a 'trial' basis (e.g. up to ten times)" and column 11, lines 1-9, "Upon credit approval, the sales representative gives the customer a special code number(s) that 'unlocks' the software products(s) for unrestricted use");

e) a programmable address module which can be programmed with an identification code uniquely associated with the LCS (column 7, lines 43-57, "a watermark or copyright notice that is inserted into the original material" and column 9, lines 43-52);

said domain processor permitting the LCS to deliver digital content to and receive digital content from an SU that is connected to the LCS's interface, provided the LCS first determines that the digital content being delivered to the SU is authorized for use by the SU or that the digital content being received is authorized for use by the LCS, and if the digital content is not authorized for use, accepting the digital content at a predetermined quality level, said predetermined quality level having been set for legacy content (column 5, lines 61-64, column 7, lines 22-57, "provides a secure system which limits unauthorized access to the materials" and column 13, lines 10-58, "denatured audio that is of adequate quality for evaluation purposes, but not for regular listening" and "VCA drops the amplitude of the source audio signal by 20 dB for a series of 20 millisecond intervals"),

said domain processor permitting the LCS to receive digital content from an SECD that is connected to the LCS's communication port, provided the LCS first determines that digital content being received is authorized for use by the LCS <u>and if the digital content is not authorized for use by the LCS, accepting the digital content at a predetermined quality level, said predetermined quality level having been set for legacy content</u> (column 5, lines 61-64, column 7, lines 22-57, "provides a secure system which limits unauthorized access to the materials" and column 13, lines 10-58, "denatured audio that is of adequate quality for evaluation purposes, but not for regular listening" and "VCA drops the amplitude of the source audio signal by 20 dB for a series of 20 millisecond intervals").

6. Regarding claim 4, Stringer discloses wherein said domain processor determines whether digital content is authorized for use by extracting a watermark from the digital content being transferred (column 7, lines 43-57, "a watermark or copyright notice that is inserted into the original material" and column 9, lines 43-52).

7. Regarding claim 5, Stringer discloses wherein said domain processor comprises:

means for obtaining identification code from an SU connected to the LCS's interface (column 7, lines 43-57, "a watermark or copyright notice that is inserted into the original material" and column 9, lines 43-52);

an analyzer to analyze the identification code from the SU to determine if the SU is an authorized device for communicating with the LCS (column 7, lines 22-57, "provides a secure system which limits unauthorized access to the materials", column 9, lines 43-68 and column 10, lines 1-8);

means for analyzing digital content received from an SU (column 7, lines 22-57, "provides a secure system which limits unauthorized access to the materials", column 9, lines 43-68 and column 10, lines 1-8);

said system permitting the digital content to be stored in the LCS if i) an analysis of the digital content received from the SU concludes that the content is authenticated, or ii) an analysis of the digital content received from the SU concludes that the content cannot be authenticated because no authentication data is embedded in the content (column 6, lines 61-66, "verifying an enable code", column 9, lines 53-68 and column 10, lines 1-20),

said system preventing the digital content from being stored on the LCS if i) an analysis of the digital content received from the SU concludes that the content is unauthenticated (column 6, lines 61-66, "verifying an enable code", column 9, lines 53-68, "If the code fails the verification step, the process is halted (21) and additional use of the product is disabled" and column 10, lines 1-20 and 43-52, "When the software application is run without using the present invention (in this case, process P0), the application gives an error message and terminates program operation").

8. Regarding claim 6, Stringer discloses wherein said analyzer of the domain processor comprises means for extracting digital watermarks from the digital content received from an SU, and means for analyzing the digital watermark to determine if the digital content has been previously marked with the unique identification code of the LCS (column 7, lines 43-57, "a watermark or copyright notice that is inserted into the original material").

9. Regarding claim 7, Stringer discloses wherein said system permits the digital content to be stored in the LCS at a degraded quality level if an analysis of the digital content received from the SU concludes that the digital content received from the SU cannot be authenticated because there is no authentication data embedded in the content (column 3, lines 55-61, "time-limited and/or function limited use of the data", column 4, lines 6-22, column 5, lines 41-48 and 61-64, column 6, lines 4-11, column 7, lines 22-57, "provides a secure system which limits unauthorized access to the materials", column 8, lines 39-44, "placed on a temporary medium, such as a random access memory in a computer system" and lines 63-68, column 9, lines 1-13, column 10, lines 43-52 and 60-68", lets customers work with the software on a 'trial' basis (e.g. up to ten times)", column 11, lines 1-9, "Upon credit approval, the sales representative gives the customer a special code number(s) that 'unlocks' the software products(s) for unrestricted use" and column 13, lines 10-58, "denatured audio that is of adequate quality for evaluation purposes, but not for regular listening" and "VCA drops the amplitude of the source audio signal by 20 dB for a series of 20 millisecond intervals").

10. Regarding claim 8, Stringer discloses at least one SU, each SU being capable of communicating with the LCS (column 4, lines 33-57, column 5, lines 35-40 and 53-64, column 6, lines 1-3 and 61-66, column 9, lines 43-63, "transaction code is given to a vendor sales representative at a remote location (61), e.g. over the telephone lines (65)", column 10, lines 53-68 and column 11, lines 1-9).

11. Regarding claim 9, Stringer discloses wherein the SU has means to sending a message to the LCS indicating that the SU is requesting a copy of a content data set

that is stored on the LCS, said message including information about the identity of the SU, and wherein the LCS comprises:

means to analyze the message from the SU to confirm that the SU is authorized to use the LCS (column 7, lines 22-57, "provides a secure system which limits unauthorized access to the materials", column 9, lines 43-68 and column 10, lines 1-8);

means to retrieve a copy of the requested content data set (column 4, lines 33-57, "remote transactions for delivery of the materials", column 7, lines 6-21, column 9, lies 43-68, column 10, lines 1-8 and 53-68, column 11, lines 1-32 and column 13, lines 10-35);

means to embed at least one robust open watermark into the copy of the requested content data set, said watermark indicating that the copy is authenticated (column 7, lines 43-57, "a watermark or copyright notice that is inserted into the original material");

means to embed a second watermark into the copy of the requested content data set, said second watermark being created based upon information transmitted by the SU and information about the LCS (column 7, lines 43-57, "a watermark or copyright notice that is inserted into the original material");

means to deliver the watermarked content data set to the SU for its use (column 7, lines 43-57, "a watermark or copyright notice that is inserted into the original material").

12. Regarding claim 10, Stringer discloses a SECD, said SECD capable of receiving a request to transfer at least one data set and capable of transmitting the at least one data set in a secured transmission (column 4, lines 33-57, "remote transactions for delivery of the materials", column 7, lines 6-57, "provides a secure system which limits unauthorized access to the materials", column 9, lies 43-68, column 10, lines 1-8 and 53-68, column 11, lines 1-32 and column 13, lines 10-35).

13. Regarding claim 11, Stringer discloses wherein the SU includes means to send a message to the LCS indicating that the SU is requesting a copy of a content data set that is not stored on the LCS, but which the LCS can obtain from an SECD, said message including information about the identity of the SU (column 8, lines 39-44, "placed on a temporary medium, such as a random access memory in a computer system");

wherein the SECD comprises:

means to retrieve a copy of the requested content data set (column 4, lines 33-57, "remote transactions for delivery of the materials", column 7, lines 6-21, column 9, lies 43-68, column 10, lines 1-8 and 53-68, column 11, lines 1-32 and column 13, lines 10-35);

means to embed at least one robust open watermark into the copy of the requested content data set, said watermark indicating that the copy is authenticated (column 7, lines 43-57, "a watermark or copyright notice that is inserted into the original material");

means to embed a second watermark into the copy of the requested content data set, said second watermark being created based upon information transmitted by the LCS (column 7, lines 43-57, "a watermark or copyright notice that is inserted into the original material");

means to deliver the watermarked content data set to the LCS for its use (column 7, lines 43-57, "a watermark or copyright notice that is inserted into the original material");

wherein the LCS comprises:

means to analyze the message from the SU to confirm that the SU is authorized to use the LCS (column 7, lines 22-57, "provides a secure system which limits unauthorized access to the materials", column 9, lines 43-68 and column 10, lines 1-8);

means to receive a copy of the requested content data set as transmitted by the SECD (column 4, lines 33-57, "remote transactions for delivery of the materials", column 7, lines 6-21, column 9, lies 43-68, column 10, lines 1-8 and 53-68, column 11, lines 1-32 and column 13, lines 10-35);

means to extract at least one robust open watermark to confirm that the content data is authorized for use by the LCS (column 7, lines 43-57, "a watermark or copyright notice that is inserted into the original material");

means to embed at least one robust open watermark into the copy of the requested content data set, said watermark indicating that the copy is authenticated (column 6, lines 61-66, "verifying an enable code", column 7, lines

43-57, "a watermark or copyright notice that is inserted into the original material", column 9, lines 53-68 and column 10, lines 1-20);

means to embed a second watermark into the copy of the requested content data set, said second watermark being created based upon information transmitted by the SU and information about the LCS (column 7, lines 43-57, "a watermark or copyright notice that is inserted into the original material");

means to deliver the watermarked content data set to the SU for its use (column 7, lines 43-57, "a watermark or copyright notice that is inserted into the original material").

14. Regarding claim 12, Stringer discloses wherein the SU has means to sending a message to the LCS indicating that the SU is requesting to store a copy of a content data set on a storage unit of the LCS, said message including information about the identity of the SU, and wherein the LCS comprises:

means to analyze the message from the SU to confirm that the SU is authorized to use the LCS (column 4, lines 33-57, column 7, lines 22-33, "provides a secure system which limits unauthorized access to the materials" and column 9, lines 43-67);

means receive a copy of the content data set (column 4, lines 33-57, "remote transactions for delivery of the materials", column 7, lines 6-21, column 9, lies 43-68, column 10, lines 1-8 and 53-68, column 11, lines 1-32 and column 13, lines 10-35);

means to determine if a robust open watermark is embedded in the content data set, and to extract the robust open watermark if it is determined that one exists (column 7, lines 43-57, "a watermark or copyright notice that is inserted into the original material");

means to analyze any extracted robust open watermarks to determine if the content data set can be authenticated (column 7, lines 43-57, "a watermark or copyright notice that is inserted into the original material");

means to permit the storage of the content data set on a storage unit of the LCS if i) the LCS authenticates the content data set, or ii) the LCS determines that no robust open watermark is embedded in the content signal (column 3, lines 55-61, "time-limited and/or function limited use of the data", column 4, lines 6-22, column 5, lines 41-48 and 61-64, column 6, lines 4-11 and 61-66, "verifying an enable code", column 7, lines 22-57, , "provides a secure system which limits unauthorized access to the materials" and "a watermark or copyright notice that is inserted into the original material", column 8, lines 39-44, "placed on a temporary medium, such as a random access memory in a computer system" and lines 63-68, column 9, lines 1-13 and 53-68 and column 10, lines 1-20, 43-52 and 60-68, "lets customers work with the software on a 'trial' basis (e.g. up to ten times)").

15. Regarding claim 13, Stringer discloses at least one SU, each such SU being capable of communicating with the LCS, and being capable of using only data which has been authorized for use by the SU or which has been determined to be legacy

content such that the data contains no additional information to permit authentication (column 3, lines 55-61, "time-limited and/or function limited use of the data", column 4, lines 6-22, column 5, lines 41-48 and 61-64, column 6, lines 4-11 and 61-66, "verifying an enable code", column 7, lines 22-57, "provides a secure system which limits unauthorized access to the materials" and "a watermark or copyright notice that is inserted into the original material", column 8, lines 39-44, "placed on a temporary medium, such as a random access memory in a computer system" and lines 63-68, column 9, lines 1-13 and 53-68 and column 10, lines 1-20, 43-52 and 60-68, "lets customers work with the software on a 'trial' basis (e.g. up to ten times)").

16. Regarding claim 14, Stringer discloses wherein the LCS further comprises:

means to embed at least one robust open watermark into a copy of content data, said watermark indicating that the copy is authenticated (column 7, lines 43-57, "a watermark or copyright notice that is inserted into the original material");

means to embed a second watermark into the copy of content data, said second watermark being created based upon information comprising information uniquely associated with the LCS (column 7, lines 43-57, "a watermark or copyright notice that is inserted into the original material");

means to embed a third watermark into the copy of content data, said third watermark being a fragile watermark created based upon information which can enhance the use of the content data on one or more SUs (column 7, lines 43-57, "a watermark or copyright notice that is inserted into the original material").

17. Regarding claim 15, Stringer discloses wherein the LCS further comprises:

means for encrypting or scrambling content data, such that content data may be encrypted or scrambled before it is stored in the rewritable storage medium (column 2, lines 65-68, column 3, lines 1-5, column 5, lines 26-32, column 6, lines 4-11 and 17-33, column 9, lines 14-24 and 43-52 and column 11, lines 33-37).

18. Regarding claim 16, Stringer discloses a system for creating a secure environment for digital content, comprising:

a Secure Electronic Content Distributor (SECD) (column 3, lines 25-30, "floppy diskette copy protection", column 4, lines 49-57, column 5, lines 35-40 and 53-60, column 9, lines 53-63, "transaction code is given to a vendor sales representative at a remote location" and column 12, lines 13-59);

a Local Content Server (LCS) (column 8, lines 39-44, "placed on a temporary medium, such as a random access memory in a computer system");

a communications network interconnecting the SECD to the LCS (column 4, lines 33-57, column 5, lines 35-40 and 53-64, column 6, lines 1-3 and 61-66, column 9, lines 43-63, "transaction code is given to a vendor sales representative at a remote location (61), e.g. over the telephone lines (65)", column 10, lines 53-68 and column 11, lines 1-9);

a Satellite Unit (SU) capable of interfacing with the LCS (column 4, lines 33-57, column 5, lines 35-40 and 53-64, column 6, lines 1-3 and 61-66, column 9, lines 43-63, "transaction code is given to a vendor sales representative at a remote location (61), e.g. over the telephone lines (65)", column 10, lines 53-68 and column 11, lines 1-9);

said SECD comprising:

a storage device for storing a plurality of data sets (column 8, lines 39-44, "placed on a temporary medium, such as a random access memory in a computer system and column 10, lines 53-59);

an input for receiving a request from the LCS to purchase a selection of at least one of said plurality of data sets (column 4, lines 33-57, column 7, lines 22-33, column 10, lines 60-68, column 11, lines 1-25 and column 12, lines 4-12 and 40-59);

a transaction processor for validating the request to purchase and for processing payment for the request (column 4, lines 33-57, column 7, lines 22-33, column 10, lines 60-68, column 11, lines 1-25 and column 12, lines 4-12 and 40-59);

a security module for encrypting or otherwise securing the selected at least one data set (column 2, lines 65-68, column 3, lines 1-5, column 5, lines 26-32, column 6, lines 4-11 and 17-33, column 9, lines 14-24 and 43-52 and column 11, lines 33-37);

an output for transmitting the selected at least one data set that has been encrypted or otherwise secured for transmission over the communications network to the LCS (column 5, lines 26-32, column 6, lines 4-11 and 17-33, column 9, lines 14-24 and 43-52 and column 11, lines 33-37);

said LCS comprising:

a domain processor (column 10, lines 60-68", lets customers work with the software on a 'trial' basis (e.g. up to ten times)");

a first interface for connecting to a communications network (column 4, lines 33-57, column 5, lines 35-40 and 53-64, column 6, lines 1-3 and 61-66, column 9, lines 43-

63, "transaction code is given to a vendor sales representative at a remote location (61), e.g. over the telephone lines (65)", column 10, lines 53-68 and column 11, lines 1-9);

a second interface for communicating with the SU (column 4, lines 33-57, column 5, lines 35-40 and 53-64, column 6, lines 1-3 and 61-66, column 9, lines 43-63, "transaction code is given to a vendor sales representative at a remote location (61), e.g. over the telephone lines (65)", column 10, lines 53-68 and column 11, lines 1-9);

a memory device for storing a plurality of data sets (column 8, lines 39-44, "placed on a temporary medium, such as a random access memory in a computer system");

a programmable address module which can be programmed with an identification code uniquely associated with the LCS (column 7, lines 43-57, "a watermark or copyright notice that is inserted into the original material" and column 9, lines 43-52);

said SU being a portable medium comprising:

a memory for accepting secure digital content from a LCS, said digital content comprising data which can be authorized for use or which has been determined to be legacy content such that the data contains no additional information to permit authentication (column 6, lines 61-66, "verifying an enable code", column 7, lines 22-57, "provides a secure system which limits unauthorized access to the materials", column 9, lines 53-68, "If the code fails the verification step, the process is halted (21) and additional use of the product is disabled" and column 10, lines 1-20 and 43-52, "When the software application is run without using the present invention (in this case, process P0), the application gives an error message and terminates program operation");

an interface for communicating with the LCS (column 4, lines 33-57, column 5, lines 35-40 and 53-64, column 6, lines 1-3 and 61-66, column 9, lines 43-63, "transaction code is given to a vendor sales representative at a remote location (61), e.g. over the telephone lines (65)", column 10, lines 53-68, column 11, lines 1-9, column 12, lines 4-63);

a programmable address module which can be programmed with an identification code uniquely associated with the SU (column 7, lines 43-57, "a watermark or copyright notice that is inserted into the original material" and column 9, lines 43-52).

19. Regarding claim 17, Stringer teaches a method for creating a secure environment for digital content for a consumer, comprising the following steps:

sending a message indicating that a user is requesting a copy of a content data set (column 9, lines 53-63, "transaction code is given to a vendor sales representative at a remote location" and column 12, lines 3-59);

retrieving a copy of the requested content data set (column 9, lines 53-63, "transaction code is given to a vendor sales representative at a remote location" and column 12, lines 3-59);

embedding at least one robust open watermark into the copy of the requested content data set, said watermark indicating that the copy is authenticated (column 6, lines 61-66, "verifying an enable code", column 7, lines 43-57, "a watermark or copyright notice that is inserted into the original material", column 9, lines 43-68 and column 10, lines 1-20);

embedding a second watermark into the copy of the requested content data set, said second watermark being created based upon information transmitted by the requesting user (column 6, lines 61-66, "verifying an enable code", column 7, lines 43-57, "a watermark or copyright notice that is inserted into the original material", column 9, lines 43-68 and column 10, lines 1-20);

transmitting the watermarked content data set into a Local Content Server (LCS) of the user (column 6, lines 61-66, "verifying an enable code", column 7, lines 43-57, "a watermark or copyright notice that is inserted into the original material", column 9, lines 43-68 and column 10, lines 1-20);

receiving the transmitted watermarked content data set into a Local Content Server (LCS) of the user (column 6, lines 61-66, "verifying an enable code", column 7, lines 43-57, "a watermark or copyright notice that is inserted into the original material", column 9, lines 43-68 and column 10, lines 1-20);

extracting at least one watermark from the transmitted watermarked content data set (column 6, lines 61-66, "verifying an enable code", column 7, lines 43-57, "a watermark or copyright notice that is inserted into the original material", column 9, lines 43-68 and column 10, lines 1-20);

permitting use of the content data set if the LCS determines that use is authorized (column 7, lines 22-57, "provides a secure system which limits unauthorized access to the materials" and column 11, lines 1-9, "Upon credit approval, the sales representative gives the customer a special code number(s) that 'unlocks' the software products(s) for unrestricted use");

permitting use of the content data set at a predetermined quality level, said predetermined quality level has been set for legacy content if the LCS determines that use is not authorized (column 5, lines 61-64, column 7, lines 22-57, "provides a secure system which limits unauthorized access to the materials" and column 13, lines 10-58, "denatured audio that is of adequate quality for evaluation purposes, but not for regular listening" and "VCA drops the amplitude of the source audio signal by 20 dB for a series of 20 millisecond intervals").

20. Regarding claim 18, Stringer teaches wherein the step of permitting use of the content data set if the LCS determines that use is authorized comprises:

checking to see if a watermark extracted from the content data set includes information which matches unique information which is associated with the user (column 6, lines 61-66, "verifying an enable code", column 7, lines 43-57, "a watermark or copyright notice that is inserted into the original material", column 9, lines 43-68 and column 10, lines 1-20);

permitting the storage of the content data set in a storage unit for the LCS (column 8, lines 39-44, "placed on a temporary medium, such as a random access memory in a computer system").

21. Regarding claim 19, Stringer teaches connecting a Satellite Unit (SU) to an LCS, wherein the step of permitting use of the content data set if the LCS determines that use is authorized comprises:

checking to see if a watermark extracted from the content data set includes information which matches unique information which is associated with the user (column 6, lines 61-66, "verifying an enable code", column 7, lines 43-57, "a watermark or copyright notice that is inserted into the original material", column 9, lines 43-68 and column 10, lines 1-20);

embedding a watermark into the content data set using information that is associated with the user and information that is associated with an SU (column 6, lines 61-66, "verifying an enable code", column 7, lines 43-57, "a watermark or copyright notice that is inserted into the original material", column 9, lines 43-68 and column 10, lines 1-20);

delivering the content data set to the SU for its use (column 4, lines 33-57, column 5, lines 35-40 and 53-64, column 6, lines 1-3 and 61-66, column 9, lines 43-63, "transaction code is given to a vendor sales representative at a remote location (61), e.g. over the telephone lines (65)", column 10, lines 53-68 and column 11, lines 1-9).

22. Regarding claim 20, Stringer teaches a method for creating a secure environment for digital content for a consumer, comprising the following steps: connecting a Satellite Unit to a local content server (LCS) (column 4, lines 33-57, column 5, lines 35-40 and 53-64, column 6, lines 1-3 and 61-66, column 9, lines 43-63, "transaction code is given to a vendor sales representative at a remote location (61), e.g. over the telephone lines (65)", column 10, lines 53-68 and column 11, lines 1-9), sending a message indicating that the SU is requesting a copy of a content data set that is stored on the LCS, said message including information about the identity of the SU (column 9, lines 53-63, "transaction code is given to a vendor sales representative at a remote location" and column 12, lines 3-59);

analyzing the message to confirm that the SU is authorized to use the LCS (column 7, lines 22-57, "provides a secure system which limits unauthorized access to the materials", column 9, lines 43-68 and column 10, lines 1-8);

retrieving a copy of the requested content data set (column 3, lines 55-61, "time-limited and/or function limited use of the data", column 4, lines 6-22, column 5, lines 41-48, column 6, lines 4-11, column 8, lines 39-44 and 63-68, column 9, lines 1-13, column 10, lines 60-68", lets customers work with the software on a 'trial' basis (e.g. up to ten times)" and column 11, lines 1-9, "Upon credit approval, the sales representative gives the customer a special code number(s) that 'unlocks' the software products(s) for unrestricted use");

assessing whether a secured connection exists between the LCS and the SU (column 6, lines 61-66, "verifying an enable code", column 7, lines 22-57, "provides a secure system which limits unauthorized access to the materials", column 9, lines 53-68, "If the code fails the verification step, the process is halted (21) and additional use of the product is disabled" and column 10, lines 1-20 and 43-52, "When the software application is run without using the present invention (in this case, process P0), the application gives an error message and terminates program operation");

if a secured connection exists, embedding a watermark into the copy of the requested content data set, said watermark being created based upon information transmitted by the SU and information about the LCS (column 6, lines 61-66, "verifying an enable code", column 7, lines 22-57, "a watermark or copyright notice that is inserted into the original material", column 9, lines 43-68 and column 10, lines 1-20);

delivering the content data set to the SU for its use, said content data set delivered at a predetermined quality level, said predetermined quality level having been set for legacy content if the LCS determines that use is not authorized (column 5, lines 61-64, column 7, lines 22-57, "provides a secure system which limits unauthorized access to the materials" and column 13, lines 10-58, "denatured audio that is of adequate quality for evaluation purposes, but not for regular listening" and "VCA drops the amplitude of the source audio signal by 20 dB for a series of 20 millisecond intervals").

23. Regarding claim 21, Stringer teaches embedding an open watermark into the content data to permit enhanced usage of the content data by the user (column 7, lines 22-57, "a watermark or copyright notice that is inserted into the original material", column 11, lines 1-9, "Upon credit approval, the sales representative gives the customer a special code number(s) that 'unlocks' the software products(s) for unrestricted use").

24. Regarding claim 22, Stringer teaches embedding at least one additional watermark into the content data (column 7, lines 43-57, "a watermark or copyright notice that is inserted into the original material" and column 9, lines 43-52);

said at least one additional watermark being based on information about the user, the LCS and an origin of the content data, said watermark serving as a forensic watermark to permit forensic analysis to provide information on the history of the content data's use (column 7, lines 43-57, "a watermark or copyright notice that is inserted into the original material" and column 9, lines 43-52);.

25. Regarding claim 23, Stringer teaches wherein the content data can be stored at a level of quality which is selected by a user (column 11, lines 2-15, "Upon credit

approval, the sales representative gives the customer a special code number(s) that 'unlocks' the software products(s) for unrestricted use").

26. Regarding claim 24, Stringer teaches a method for creating a secure environment for digital content for a consumer, comprising the following steps:

connecting a Satellite Unit (SU) to a local content server (LCS) (column 4, lines 33-57, column 5, lines 35-40 and 53-64, column 6, lines 1-3 and 61-66, column 9, lines 43-63, "transaction code is given to a vendor sales representative at a remote location (61), e.g. over the telephone lines (65)", column 10, lines 53-68 and column 11, lines 1-9),

sending a message indicating that the SU is requesting a copy of a content data set that is stored on the LCS, said message including information about the identity of the SU (column 9, lines 53-63, "transaction code is given to a vendor sales representative at a remote location" and column 12, lines 3-59);

analyzing the message to confirm that the SU is authorized to use the LCS (column 7, lines 22-57, "provides a secure system which limits unauthorized access to the materials", column 9, lines 43-68 and column 10, lines 1-8);

retrieving a copy of the requested content data set (column 3, lines 55-61, "time-limited and/or function limited use of the data", column 4, lines 6-22, column 5, lines 41-48, column 6, lines 4-11, column 8, lines 39-44 and 63-68, column 9, lines 1-13, column 10, lines 60-68", lets customers work with the software on a 'trial' basis (e.g. up to ten times)" and column 11, lines 1-9, "Upon credit approval, the sales representative gives the customer a special code number(s) that 'unlocks' the software products(s) for unrestricted use");

assessing whether a secured connection exists between the LCS and the SU (column 6, lines 61-66, "verifying an enable code", column 7, lines 22-57, "provides a secure system which limits unauthorized access to the materials", column 9, lines 53-68, "If the code fails the verification step, the process is halted (21) and additional use of the product is disabled" and column 10, lines 1-20 and 43-52, "When the software application is run without using the present invention (in this case, process P0), the application gives an error message and terminates program operation");

if a secured connection exists, embedding a watermark into the copy of the requested content data set, said watermark being created based upon information transmitted by the SU and information about the LCS (column 6, lines 61-66, "verifying an enable code", column 7, lines 22-57, "a watermark or copyright notice that is inserted into the original material", column 9, lines 43-68 and column 10, lines 1-20);

delivering the watermarked content data set to the SU for its use, said watermarked content data set delivered at a predetermined quality level, said predetermined quality having been set for legacy content if the LCS determines that use is not authorized (column 5, lines 61-64, column 7, lines 22-57, "provides a secure system which limits unauthorized access to the materials" and column 13, lines 10-58, "denatured audio that is of adequate quality for evaluation purposes, but not for regular listening" and "VCA drops the amplitude of the source audio signal by 20 dB for a series of 20 millisecond intervals").

27. Regarding claim 25, Stringer teaches embedding at least one robust open watermark into the copy of the requested content data set before the requested content

data is delivered to the SU, said watermark indicating that the copy is authenticated (column 7, lines 22-57, "a watermark or copyright notice that is inserted into the original material").

28. Regarding claim 26, Stringer teaches wherein the robust watermark is embedded using any one of a plurality of embedding algorithms (column 7, lines 43-57, "a watermark or copyright notice that is inserted into the original material" and column 9, lines 43-52).

29. Regarding claim 27, Stringer teaches embedding a watermark which includes a hash value from a one-way hash function using the content data ((column 7, lines 43-57, "a watermark or copyright notice that is inserted into the original material" and column 9, lines 14-24, "denaturing process is a unique, check-summed operation using any of the many known encryption algorithms, such as the data encryption standard published by the U.S. government ("DES")" and lines 43-52).

30. Regarding claim 28, Stringer teaches wherein the robust watermark can be periodically replaced with a new robust watermark generated using a new algorithm with payload that is no greater than that utilized by the old robust watermark (column 6, lines 52-66, "hidden portion A1", column 7, lines 43-57, "a watermark or copyright notice that is inserted into the original material" and column 9, lines 43-52).

31. Regarding claim 29, Stringer teaches embedding additional robust open watermarks into the copy of the requested content data set before the requested content data is delivered to the SU, using a new algorithm (column 6, lines 52-66,

"hidden portion A1", column 7, lines 43-57, "a watermark or copyright notice that is inserted into the original material" and column 9, lines 43-52);

re-saving the newly watermarked copy to the LCS (column 6, lines 52-66, "hidden portion A1", column 7, lines 43-57, "a watermark or copyright notice that is inserted into the original material" and column 9, lines 43-52).

32. Regarding claim 30, Stringer teaches saving a copy of the requested content data with the robust watermark to the rewritable media of the LCS (column 6, lines 52-66, "hidden portion A1", column 7, lines 43-57, "a watermark or copyright notice that is inserted into the original material", column 8, lines 39-44, "placed on a temporary medium, such as a random access memory in a computer system" and column 9, lines 43-52).

33. Regarding claim 31, Stringer teaches a method of creating a secure environment for digital content for a consumer, comprising the following steps:

connecting a Satellite Unit (SU) to a local content server (LCS) (column 4, lines 33-57, column 5, lines 35-40 and 53-64, column 6, lines 1-3 and 61-66, column 9, lines 43-63, "transaction code is given to a vendor sales representative at a remote location (61), e.g. over the telephone lines (65)", column 10, lines 53-68 and column 11, lines 1-9), sending a message indicating that the SU is requesting a copy of a content data set that is stored on the LCS, said message including information about the identity of the SU (column 9, lines 53-63, "transaction code is given to a vendor sales representative at a remote location" and column 12, lines 3-59), sending a message indicating that the SU is requesting to store a copy of a content data on the LCS, said message including information about the identity of the SU (column 8, lines 39-44, "placed on a temporary medium, such as a random access memory in a computer system", column 9, lines 53-63, "transaction code is given to a vendor sales representative at a remote location" and column 12, lines 3-59);

analyzing the message to confirm that the SU is authorized to use the LCS (column 7, lines 22-57, "provides a secure system which limits unauthorized access to the materials", column 9, lines 43-68 and column 10, lines 1-8);

receiving a copy of the content data set (column 3, lines 55-61, "time-limited and/or function limited use of the data", column 4, lines 6-22, column 5, lines 41-48, column 6, lines 4-11, column 8, lines 39-44 and 63-68, column 9, lines 1-13, column 10, lines 60-68", lets customers work with the software on a 'trial' basis (e.g. up to ten times)" and column 11, lines 1-9, "Upon credit approval, the sales representative gives the customer a special code number(s) that 'unlocks' the software products(s) for unrestricted use");

assessing whether the content data is authenticated (column 6, lines 61-66, "verifying an enable code", column 9, lines 53-68, "If the code fails the verification step, the process is halted (21) and additional use of the product is disabled" and column 10, lines 1-20 and 43-52, "When the software application is run without using the present invention (in this case, process P0), the application gives an error message and terminates program operation");

if the content data is unauthenticated, denying access to the LCS storage unit (column 6, lines 61-66, "verifying an enable code", column 9, lines 53-68, "If the code fails the verification step, the process is halted (21) and additional use of the product is disabled" and column 10, lines 1-20 and 43-52, "When the software application is run without using the present invention (in this case, process P0), the application gives an error message and terminates program operation");

if the content data is not capable of authentication, accepting the data at a predetermined quality level, said predetermined quality level having been set for legacy content (column 5, lines 61-64, column 7, lines 22-57, "provides a secure system which limits unauthorized access to the materials" and column 13, lines 10-58, "denatured audio that is of adequate quality for evaluation purposes, but not for regular listening" and "VCA drops the amplitude of the source audio signal by 20 dB for a series of 20 millisecond intervals").

### ***Response to Arguments***

34. Applicant's arguments, see pages 15-19, filed 7/3/06, with respect to the rejection(s) of claim(s) 1-13 and 15-31 under 35 U.S.C. 102(e) have been fully considered and are persuasive. Therefore, the rejection has been withdrawn. However, upon further consideration, a new ground(s) of rejection is made in view of newly found prior art.

35. Applicant's arguments, see pages 19 and 20, filed 7/3/06, with respect to the rejection(s) of claim(s) 14 under 35 U.S.C. 103(a) have been fully considered and are persuasive. Therefore, the rejection has been withdrawn. However, upon further consideration, a new ground(s) of rejection is made in view of newly found prior art.

### ***Conclusion***

36. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

37. The following United States Patents are cited to further show the state of the art with respect to secure delivery of content, such as:

United States Patent No. 6,966,002 to Torrubia-Saez which is cited to show methods and apparatus for secure distribution of software.

United States Patent No. 6,263,313 to Milsted et al., which is cited to show a method and apparatus to create encoded digital content.

United States Patent No. 7,093,295 to Saito which is cited to show a method and device for protecting digital data by double re-encryption.

United States Patent No. 6,587,837 to Spagna et al., which is cited to show a method for delivering content from an online store.

United States Patent No. 6,931,534 to Jandel et al., which is cited to show a method and a device for encryption of images.

38. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

39. A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the

shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

40. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jeremiah Avery whose telephone number is (571) 272-8627. The examiner can normally be reached on Monday thru Friday 8:30am-5pm.

41. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ayaz Sheikh can be reached on (571) 272-3795. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

42. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

JLA

AYAZ SHEIKH
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

OIPE IAP7A
JUL 0 3 2006
PATENT & TRADEMARK OFFICE

## IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appl. No. : 10/049,101 Confirmation No. 8028
Applicant : Scott A. MOSKOWITZ
Filed : July 22, 2002
TC/A.U. : 2131
Examiner : AVERY, Jeremiah L.

Docket No. : 80408.0011

**MAIL STOP AMENDMENT**
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

## INFORMATION DISCLOSURE STATEMENT

Dear Sir:

Applicants submit copies of the references listed on the attached SB08 Form for consideration and request that the U.S. Patent and Trademark Office make them of record in this application.

Applicants state the following:

☐ Each item of information contained in this Information Disclosure Statement was cited in a communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of the Information Disclosure Statement; or

☐ No item of information contained in this Information Disclosure Statement was cited in a communication from a foreign patent office in a counterpart foreign application, and to the knowledge of Applicant(s) no item of information contained in this Information Disclosure Statement was known to any individual designated in § 1.56(c) more than three months prior to the filing of this Information Disclosure Statement.

07/07/2006 HDESTA1 00000040 10049101
01 FC:1806 180.00 DP

☐ In accordance with 37 C.F.R. § 1.97(b), this Information Disclosure Statement is believed to be submitted prior to issuance of a first Office Action and/or within three months of the filing date of the application. It is respectfully submitted that no fee is required for consideration of this information.

☒ This Information Disclosure Statement is being submitted after the mailing of a non-final Office Action, but is believed to be prior to a final Office Action or a Notice of Allowance. Pursuant to 37 C.F.R. § 1.97(c), payment in the amount of $180.00 as set forth in 37 C.F.R. § 1.17(p) is enclosed.

While the information and references disclosed in this Information Disclosure Statement are submitted pursuant to 37 C.F.R. § 1.56, this submission is not intended to constitute an admission that any patent, publication or other information referred to is "prior art" to this invention. Applicants reserve the right to contest the "prior art" status of any information submitted or asserted against the application.

Additionally, Applicant wishes to inform the Examiner of the existence of the following co-pending U.S. patents and patent applications that share a common inventor with the present application:

EXAMINER'S INITIALS:

/JA/ U.S. Patent Application No. 08/999,766, filed July 23, 1997, entitled "Steganographic Method and Device";

C ~~EPO Application No. 96919405.9, entitled "Steganographic Method and Device";~~

/JA/ U.S. Patent Application No. 08/674,726, filed July 2, 1996, entitled "Exchange Mechanisms for Digital Information Packages with Bandwidth Securitization, Multichannel Digital Watermarks, and Key Management";

/JA/ U.S. Patent Application No. 09/545,589, filed April 7, 2000, entitled "Method and System for Digital Watermarking";

/JA/ U.S. Patent Application No. 09/046,627, filed March 24, 1998, entitled "Method for Combining Transfer Function with Predetermined Key Creation" now U.S. Patent No. 6,598,162, July, 22, 2003;

/JA/ U.S. Patent Application No. 09/053,628, filed April 2, 1998, entitled "Multiple Transform Utilization and Application for Secure Digital Watermarking";

/JA/ U.S. Patent Application No. 09/644,098, filed August 23, 2000, entitled "Multiple Transform Utilization and Application for Secure Digital Watermarking";

~~Jap. App. No.2000-542907, entitled "Multiple Transform Utilization and Application for Secure Digital Watermarking";~~

/JA/ U.S. Patent Application No. 09/767,733, filed January 24, 2001, entitled "Multiple Transform Utilization and Application for Secure Digital Watermarking";

/JA/ U.S. Patent Application No. 10/417,231, filed April 17, 2003, entitled "Methods, Systems And Devices For Packet Watermarking And Efficient Provisioning Of Bandwidth";

/JA/ U.S. Patent Application 10/602,777, filed June 25, 2003, entitled "Method for Combining Transfer Function with Predetermined Key Creation";

/JA/ U.S. Patent Application No. 10/369,344, filed February 18, 2003, entitled "Optimization Methods for the Insertion, Protection, and Detection of Digital Watermarks in Digital Data";

/JA/ U.S. Patent Application No. 09/789,711, filed Feb. 22, 2001, entitled "Optimization Methods for the Insertion, Protection, and Detection of Digital Watermarks in Digital Data";

/JA/ U.S. Patent Application No.09/594,719, filed June 16, 2000, entitled "Utilizing Data Reduction in Steganographic and Cryptographic Systems";

/JA/ U.S. Application No 09/731,040, filed December 7, 2000, entitled "Systems, Methods And Devices For Trusted Transactions";

/JA/ U.S. Patent Application No. 10/049,101, filed Feb. 8, 2002, entitled "A Secure Personal Content Server" (which claims priority to International Application No. PCT/US00/21189, filed August 4, 2000, which claims priority to U.S. Patent Application No. 60/147,134, filed August 4, 1999, and to U.S. Patent Application No. 60/213,489, filed June 23, 2000);

/JA/ PCT Application No. PCT/US00/21189, filed August 4, 2000, entitled, "A Secure Personal Content Server";

/JA/ U.S. Patent Application No. 09/657,181, filed 09/07/00, entitled "Method And Device For Monitoring And Analyzing Signals"

/JA/ U.S. Patent Application No. 10/805,484, filed 03/22/04, entitled "Method And Device For Monitoring And Analyzing Signals"(which claims priority to U.S. Patent Application No. 09/671,739, filed 09/29/00, which is a CIP of U.S. Patent Application No. 09/657,181);

/JA/ U.S. Patent Application No. 09/956,262, filed 09/20/01, entitled "Improved Security Based on Subliminal and Supraliminal Channels For Data Objects"

/JA/ U.S. Patent Application No. 11/026,234, filed December 30, 2004, entitled "Z-Transform Implementation of Digital Watermarks";

/JA/ U.S. Patent No. 5,822,432, issued October 13, 1998, entitled "Method for Human Assisted Random Key Generation ...";

/JA/ U.S. Patent No. 5,905,800, issued May 18, 1999, entitled "Method & System for Digital Watermarking";

/JA/ U.S. Patent No. 5,613,004, issued March 18, 1997, entitled "Steganographic Method and Device";

/JA/ U.S. Patent No. 5,687,236, issued November 11, 1997, entitled "Steganographic Method and Device";

/JA/ U.S. Patent No. 5,745,569, issued April 28, 1998, entitled "Method for Stega-Protection of Computer Code";

/JA/ U.S. Patent No. 6,078,664, issued June 20, 2000, entitled "Z-Transform Implementation of Digital Watermarks";

/JA/ U.S. Patent No. 6,853,726, issued February 8, 2005, entitled "Z-Transform Implementation of Digital Watermarks";

/JA/ U.S. Patent No. 5,428,606, issued June 27, 1995, entitled "Digital Commodities Exchange";

/JA/ U.S. Patent No. 5,539,735, issued July 23, 1996, entitled "Digital Information Commodities Exchange";

/JA/ U.S. Patent No. 5,889,868, issued July 2, 1996, entitled "Optimization Methods for the Insertion, Protection and Detection...";

/JA/ U.S. Patent No. 6,522,767, issued February 18, 2003, entitled "Optimization Methods for the Insertion, Protection and Detection...";

/JA/ U.S. Patent No. 6,205,249, issued March 20, 2001, entitled "Multiple Transform Utilization and Application for Secure Digital Watermarking";

/JA/ U.S. Patent No. 6,598,162, issued July 22, 2003, entitled "Method for Combining Transfer Function with Predetermined Key Creation";

/JA/ U.S. Patent No. 7,007,166, issued February 28, 2006, entitled "Method & System for Digital Watermarking";

/JA/ U.S. Patent No. 7,035,049, issued April 25, 2006, entitled "Multiple Transform Utilization and Application for Secure Digital Watermarking".

In accordance with 37 C.F.R. § 1.97(g), the filing of this Information Disclosure Statement shall not be construed to mean that a search has been made or that no other material information as defined in 37 C.F.R. § 1.56(a) exists. This Information Disclosure Statement is in compliance with 37 C.F.R. § 1.98 and the Examiner is respectfully requested to consider the listed documents and information.

Respectfully submitted,

Date: July 3, 2006

By:

Scott A. Moskowitz
Tel# (305) 956-9041
Fax# (305) 956-9042

For Blue Spike, Inc.

Scott A. Moskowitz
President

OIPE
JUL 0 3 2006
IAP74
PATENT & TRADEMARK OFFICE

PTO/SB/08A (07-05)
Approved for use through 07/31/2006. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Substitute for form 1449/PTO

# INFORMATION DISCLOSURE STATEMENT BY APPLICANT

*(Use as many sheets as necessary)*

Sheet 1 of 1

**Complete if Known**

| | |
|---|---|
| Application Number | 10/049,101 |
| Filing Date | July 23, 2002 |
| First Named Inventor | Scott A. MOSKOWITZ |
| Art Unit | 2136 |
| Examiner Name | HAST, Nathan D. |
| Attorney Docket Number | 80408.0011 |

**U. S. PATENT DOCUMENTS**

| Examiner Initials* | Cite No.[1] | Document Number<br>Number-Kind Code[2 (if known)] | Publication Date MM-DD-YYYY | Name of Patentee or Applicant of Cited Document | Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear |
|---|---|---|---|---|---|
| /JA/ | | US- 5,636,292 | 06-03-1997 | Rhoads | |
| /JA/ | | US- 5,629,980 | 05-13-1997 | Stefik et al. | |
| /JA/ | | US- 5,943,422 | 08-24-1999 | Van Wie et al. | |
| /JA/ | | US- 5,636,276 | 06-03-1997 | Brugger | |
| /JA/ | | US- 5,341,429 | 08-23-1994 | Stringer | |
| | | US- | | | |
| | | US- | | | |
| | | US- | | | |
| | | US- | | | |
| | | US- | | | |
| | | US- | | | |
| | | US- | | | |
| | | US- | | | |
| | | US- | | | |
| | | US- | | | |
| | | US- | | | |
| | | US- | | | |
| | | US- | | | |
| | | US- | | | |

**FOREIGN PATENT DOCUMENTS**

| Examiner Initials* | Cite No.[1] | Foreign Patent Document<br>Country Code[3] "Number[4] "Kind Code[5] (if known) | Publication Date MM-DD-YYYY | Name of Patentee or Applicant of Cited Document | Pages, Columns, Lines, Where Relevant Passages Or Relevant Figures Appear | T[6] |
|---|---|---|---|---|---|---|
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |

| | | | |
|---|---|---|---|
| Examiner Signature | /Jeremiah Avery/ | Date Considered | 04/26/2007 |

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. [1]Applicant's unique citation designation number (optional). [2]See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3]Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4]For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5]Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [6]Applicant is to place a check mark here if English language Translation is attached.

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

*If you need assistance in completing the form, call 1-800-PTO-9199 (1-800-786-9199) and select option 2.*

|  |  |  |  |
|---|---|---|---|
| **Notice of References Cited** | Application/Control No.<br>10/049,101 | Applicant(s)/Patent Under Reexamination<br>MOSKOWITZ, SCOTT A. | |
| | Examiner<br>Jeremiah Avery | Art Unit<br>2131 | Page 1 of 1 |

U.S. PATENT DOCUMENTS

| * | | Document Number Country Code-Number-Kind Code | Date MM-YYYY | Name | Classification |
|---|---|---|---|---|---|
| * | A | US-6,966,002 | 11-2005 | Torrubia-Saez, Andres | 726/29 |
| * | B | US-6,263,313 | 07-2001 | Milsted et al. | 705/1 |
| * | C | US-7,093,295 | 08-2006 | Saito, Makoto | 726/26 |
| * | D | US-6,587,837 | 07-2003 | Spagna et al. | 705/26 |
| * | E | US-6,931,534 | 08-2005 | Jandel et al. | 713/176 |
| * | F | US-5,341,429 | 08-1994 | Stringer et al. | 705/52 |
| | G | US- | | | |
| | H | US- | | | |
| | I | US- | | | |
| | J | US- | | | |
| | K | US- | | | |
| | L | US- | | | |
| | M | US- | | | |

FOREIGN PATENT DOCUMENTS

| * | | Document Number Country Code-Number-Kind Code | Date MM-YYYY | Country | Name | Classification |
|---|---|---|---|---|---|---|
| | N | | | | | |
| | O | | | | | |
| | P | | | | | |
| | Q | | | | | |
| | R | | | | | |
| | S | | | | | |
| | T | | | | | |

NON-PATENT DOCUMENTS

| * | | Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages) |
|---|---|---|
| | U | |
| | V | |
| | W | |
| | X | |

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

U.S. Patent and Trademark Office
PTO-892 (Rev. 01-2001) Notice of References Cited Part of Paper No. 20061002

| | | |
|---|---|---|
| ***Index of Claims*** | **Application/Control No.**<br>10/049,101 | **Applicant(s)/Patent under Reexamination**<br>MOSKOWITZ, SCOTT A. |
| | **Examiner**<br>Jeremiah Avery | **Art Unit**<br>2131 |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| √ | **Rejected** | – | **(Through numeral) Cancelled** | N | **Non-Elected** | A | **Appeal** |
| = | **Allowed** | ÷ | **Restricted** | I | **Interference** | O | **Objected** |

| Claim | | Date | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Final | Original | 4/26/07 | 4/26/07 | | | | | | | |
| | 1 | √ | | | | | | | | |
| | 2 | √ | | | | | | | | |
| | 3 | √ | | | | | | | | |
| | 4 | √ | | | | | | | | |
| | 5 | √ | | | | | | | | |
| | 6 | √ | | | | | | | | |
| | 7 | √ | | | | | | | | |
| | 8 | √ | | | | | | | | |
| | 9 | √ | | | | | | | | |
| | 10 | √ | | | | | | | | |
| | 11 | √ | | | | | | | | |
| | 12 | √ | O | | | | | | | |
| | 13 | √ | | | | | | | | |
| | 14 | √ | | | | | | | | |
| | 15 | √ | | | | | | | | |
| | 16 | √ | | | | | | | | |
| | 17 | √ | | | | | | | | |
| | 18 | √ | | | | | | | | |
| | 19 | √ | | | | | | | | |
| | 20 | √ | | | | | | | | |
| | 21 | √ | | | | | | | | |
| | 22 | √ | | | | | | | | |
| | 23 | √ | | | | | | | | |
| | 24 | √ | | | | | | | | |
| | 25 | √ | | | | | | | | |
| | 26 | √ | | | | | | | | |
| | 27 | √ | | | | | | | | |
| | 28 | √ | | | | | | | | |
| | 29 | √ | | | | | | | | |
| | 30 | √ | | | | | | | | |
| | 31 | √ | | | | | | | | |
| | 32 | | | | | | | | | |
| | 33 | | | | | | | | | |
| | 34 | | | | | | | | | |
| | 35 | | | | | | | | | |
| | 36 | | | | | | | | | |
| | 37 | | | | | | | | | |
| | 38 | | | | | | | | | |
| | 39 | | | | | | | | | |
| | 40 | | | | | | | | | |
| | 41 | | | | | | | | | |
| | 42 | | | | | | | | | |
| | 43 | | | | | | | | | |
| | 44 | | | | | | | | | |
| | 45 | | | | | | | | | |
| | 46 | | | | | | | | | |
| | 47 | | | | | | | | | |
| | 48 | | | | | | | | | |
| | 49 | | | | | | | | | |
| | 50 | | | | | | | | | |

| Claim | | Date | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Final | Original | | | | | | | | | |
| | 51 | | | | | | | | | |
| | 52 | | | | | | | | | |
| | 53 | | | | | | | | | |
| | 54 | | | | | | | | | |
| | 55 | | | | | | | | | |
| | 56 | | | | | | | | | |
| | 57 | | | | | | | | | |
| | 58 | | | | | | | | | |
| | 59 | | | | | | | | | |
| | 60 | | | | | | | | | |
| | 61 | | | | | | | | | |
| | 62 | | | | | | | | | |
| | 63 | | | | | | | | | |
| | 64 | | | | | | | | | |
| | 65 | | | | | | | | | |
| | 66 | | | | | | | | | |
| | 67 | | | | | | | | | |
| | 68 | | | | | | | | | |
| | 69 | | | | | | | | | |
| | 70 | | | | | | | | | |
| | 71 | | | | | | | | | |
| | 72 | | | | | | | | | |
| | 73 | | | | | | | | | |
| | 74 | | | | | | | | | |
| | 75 | | | | | | | | | |
| | 76 | | | | | | | | | |
| | 77 | | | | | | | | | |
| | 78 | | | | | | | | | |
| | 79 | | | | | | | | | |
| | 80 | | | | | | | | | |
| | 81 | | | | | | | | | |
| | 82 | | | | | | | | | |
| | 83 | | | | | | | | | |
| | 84 | | | | | | | | | |
| | 85 | | | | | | | | | |
| | 86 | | | | | | | | | |
| | 87 | | | | | | | | | |
| | 88 | | | | | | | | | |
| | 89 | | | | | | | | | |
| | 90 | | | | | | | | | |
| | 91 | | | | | | | | | |
| | 92 | | | | | | | | | |
| | 93 | | | | | | | | | |
| | 94 | | | | | | | | | |
| | 95 | | | | | | | | | |
| | 96 | | | | | | | | | |
| | 97 | | | | | | | | | |
| | 98 | | | | | | | | | |
| | 99 | | | | | | | | | |
| | 100 | | | | | | | | | |

| Claim | | Date | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Final | Original | | | | | | | | | |
| | 101 | | | | | | | | | |
| | 102 | | | | | | | | | |
| | 103 | | | | | | | | | |
| | 104 | | | | | | | | | |
| | 105 | | | | | | | | | |
| | 106 | | | | | | | | | |
| | 107 | | | | | | | | | |
| | 108 | | | | | | | | | |
| | 109 | | | | | | | | | |
| | 110 | | | | | | | | | |
| | 111 | | | | | | | | | |
| | 112 | | | | | | | | | |
| | 113 | | | | | | | | | |
| | 114 | | | | | | | | | |
| | 115 | | | | | | | | | |
| | 116 | | | | | | | | | |
| | 117 | | | | | | | | | |
| | 118 | | | | | | | | | |
| | 119 | | | | | | | | | |
| | 120 | | | | | | | | | |
| | 121 | | | | | | | | | |
| | 122 | | | | | | | | | |
| | 123 | | | | | | | | | |
| | 124 | | | | | | | | | |
| | 125 | | | | | | | | | |
| | 126 | | | | | | | | | |
| | 127 | | | | | | | | | |
| | 128 | | | | | | | | | |
| | 129 | | | | | | | | | |
| | 130 | | | | | | | | | |
| | 131 | | | | | | | | | |
| | 132 | | | | | | | | | |
| | 133 | | | | | | | | | |
| | 134 | | | | | | | | | |
| | 135 | | | | | | | | | |
| | 136 | | | | | | | | | |
| | 137 | | | | | | | | | |
| | 138 | | | | | | | | | |
| | 139 | | | | | | | | | |
| | 140 | | | | | | | | | |
| | 141 | | | | | | | | | |
| | 142 | | | | | | | | | |
| | 143 | | | | | | | | | |
| | 144 | | | | | | | | | |
| | 145 | | | | | | | | | |
| | 146 | | | | | | | | | |
| | 147 | | | | | | | | | |
| | 148 | | | | | | | | | |
| | 149 | | | | | | | | | |
| | 150 | | | | | | | | | |

U.S. Patent and Trademark Office

Part of Paper No. 20061002

| | | |
|---|---|---|
| **Search Notes** | **Application/Control No.**<br>10/049,101 | **Applicant(s)/Patent under Reexamination**<br>MOSKOWITZ, SCOTT A. |
| | **Examiner**<br>Jeremiah Avery | **Art Unit**<br>2131 |

### SEARCHED

| Class | Subclass | Date | Examiner |
|---|---|---|---|
| none | none | 10/1/2006 | JLA |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

### SEARCH NOTES (INCLUDING SEARCH STRATEGY)

| | DATE | EXMR |
|---|---|---|
| EAST Search | 04/26/07 | JLA |
| Inventor Search | 4/26/2007 | JLA |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

### INTERFERENCE SEARCHED

| Class | Subclass | Date | Examiner |
|---|---|---|---|
| none | none | 10/1/2006 | JLA |
| | | | |
| | | | |
| | | | |

U.S. Patent and Trademark Office

Part of Paper No. 20061002

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 10/049,101 | 07/23/2002 | Scott A. Moskowitz | 80408.0011 | 8028 |

7590 07/03/2007

Scott A. Moskowitz
#2505
16711 Collins Avenue
Miami, FL 33160

| EXAMINER | |
|---|---|
| AVERY, JEREMIAH L | |

| ART UNIT | PAPER NUMBER |
|---|---|
| 2131 | |

| MAIL DATE | DELIVERY MODE |
|---|---|
| 07/03/2007 | PAPER |

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

PTOL-90A (Rev. 04/07)

|  |  |  |  |
|---|---|---|---|
| ***Interview Summary*** | **Application No.**<br>10/049,101 | **Applicant(s)**<br>MOSKOWITZ, SCOTT A. | |
| | **Examiner**<br>Jeremiah Avery | **Art Unit**<br>2131 | |

All participants (applicant, applicant's representative, PTO personnel):

(1) *Jeremiah Avery*. (3)_____.

(2) *Scott Moskowitz*. (4)_____.

Date of Interview: *28 June 2007*.

Type: a)☒ Telephonic b)☐ Video Conference
c)☐ Personal [copy given to: 1)☐ applicant 2)☐ applicant's representative]

Exhibit shown or demonstration conducted: d)☐ Yes e)☒ No.
If Yes, brief description: _____.

Claim(s) discussed: _____.

Identification of prior art discussed: *United States Patent No. 5,341,429 to Stringer et al., hereinafter Stringer*.

Agreement with respect to the claims f)☐ was reached. g)☒ was not reached. h)☐ N/A.

Substance of Interview including description of the general nature of what was agreed to if an agreement was reached, or any other comments: *Discussion of claim elements "legacy content" and "predetermined quality level" and how they differ in scope in relation to Stringer. Mr. Moskowitz further detailed the meanings of these terms, as further defined within his Specification. Upon filing a formal written response, detailing the matters discussed within the interview, the Examiner will consider the arguments presented*.

(A fuller description, if necessary, and a copy of the amendments which the examiner agreed would render the claims allowable, if available, must be attached. Also, where no copy of the amendments that would render the claims allowable is available, a summary thereof must be attached.)

THE FORMAL WRITTEN REPLY TO THE LAST OFFICE ACTION MUST INCLUDE THE SUBSTANCE OF THE INTERVIEW. (See MPEP Section 713.04). If a reply to the last Office action has already been filed, APPLICANT IS GIVEN A NON-EXTENDABLE PERIOD OF THE LONGER OF ONE MONTH OR THIRTY DAYS FROM THIS INTERVIEW DATE, OR THE MAILING DATE OF THIS INTERVIEW SUMMARY FORM, WHICHEVER IS LATER, TO FILE A STATEMENT OF THE SUBSTANCE OF THE INTERVIEW. See Summary of Record of Interview requirements on reverse side or on attached sheet.

Examiner Note: You must sign this form unless it is an Attachment to a signed Office action.

Examiner's signature, if required

U.S. Patent and Trademark Office
PTOL-413 (Rev. 04-03) Interview Summary Paper No. 20070628

JUL 0 9 2007

Appl'n No. 10/049,101
Reply to final Office Action of May 9, 2007 dated July 9, 2007

## IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appl. No. : 10/049,101 Confirmation No. 8028
Applicant : Scott A. Moskowitz, et al.
Filed : July 23, 2002
TC/A.U. : 2131
Examiner : Jeremiah AVERY

Docket No. : 80408.0011

**Mail Stop: After Final**
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

## RESPONSE TO FINAL OFFICE ACTION

In response to the final Office Action of May 9, 2007 Applicants provide the following remarks:

**Amendments to the Claims:**

Please amend the following: Claim 12. The amendment to claim 12 is being made to correct grammatical errors and is not being made for reasons of patentability. Applicants reserve the right to pursue the subject matter of the original claims in this application and in other applications. This listing of claims will replace all prior versions, and listings, of claims in the application:

**Listing of Claims:**

1. (previously presented) A local content server system (LCS) for creating a secure environment for digital content, comprising:

a) a communications port in communication for connecting the system via a network to at least one Secure Electronic Content Distributor (SECD), said SECD capable of storing a plurality of data sets, capable of receiving a request to transfer at least one content data set, and capable of transmitting the at least one content data set in a secured transmission;

b) a rewritable storage medium whereby content received from outside the LCS may be stored and retrieved;

c) a domain processor that imposes rules and procedures for content being transferred between the LCS and devices outside the LCS; and

d) a programmable address module which can be programmed with an identification code uniquely associated with the LCS; and

said domain processor permitting the LCS to receive digital content from outside the LCS provided the LCS first determines that the digital content being delivered to the LCS is authorized for use by the LCS and if the digital content is not authorized for use by the LCS, accepting the digital content at a predetermined quality level, said predetermined quality level having been set for legacy content.

2. (original) The LCS of claim 1 further comprising

e) an interface to permit the LCS to communicate with one or more Satellite Units (SU) which may be connected to the system through the interface, said SUs capable of receiving and transmitting digital content;

and wherein said domain processor permits the LCS to receive digital content from an SECD that is connected to the LCS's communication port, provided the LCS first determines that digital content being received is authorized for use by the LCS,

and wherein said domain processor permits the LCS to deliver digital content to an SU that may be connected to the LCS's interface, provided the LCS first determines that digital content being received is authorized for use by the SU.

3. (previously presented) A local content server system (LCS) for creating a secure environment for digital content, comprising:

a) a communications port in communication for connecting the system via a network to at least one Secure Electronic Content Distributor (SECD), said SECD capable of storing a plurality of data sets, capable of receiving a request to transfer at least one content data set, and capable of transmitting the at least one content data set in a secured transmission;

b) an interface to permit the LCS to communicate with one or more Satellite Units (SU) which may be connected to the system through the interface, said SUs capable of receiving and transmitting digital content; and

c) a rewritable storage medium whereby content received from an SECD and from an SU may be stored and retrieved;

d) a domain processor that imposes rules and procedures for content being transferred between the LCS and the SECD and between the LCS and the SU; and

e) a programmable address module which can be programmed with an identification code uniquely associated with the LCS;

said domain processor permitting the LCS to deliver digital content to and receive digital content from an SU that is connected to the LCS's interface, provided the LCS first determines that the digital content being delivered to the SU is authorized for use by the SU or that the digital content being received is authorized for use by the LCS, and if the digital content is not authorized for use, accepting the digital content at a predetermined quality level, said predetermined quality level having been set for legacy content,

and said domain processor permitting the LCS to receive digital content from an SECD that is connected to the LCS's communication port, provided the LCS first determines that digital content being received is authorized for use by the LCS and if the digital content is not authorized for use by the LCS, accepting the digital content at a predetermined quality level, said predetermined quality level having been set for legacy content.

4. (original) The system of claim 3, wherein said domain processor determines whether digital content is authorized for use by extracting a watermark from the digital content being transferred.

5. (original) The system of claim 3, wherein said domain processor comprises:

means for obtaining an identification code from an SU connected to the LCS's interface;

an analyzer to analyze the identification code from the SU to determine if the SU is an authorized device for communicating with the LCS;

means for analyzing digital content received from an SU;

said system permitting the digital content to be stored in the LCS if i) an analysis of the digital content received from the SU concludes that the content is authenticated, or ii) an analysis of the digital content received from the SU concludes that the content cannot be authenticated because no authentication data is embedded in the content, and

said system preventing the digital content from being stored on the LCS if i) an analysis of the digital content received from the SU concludes that the content is unauthenticated.

6. (original) The system of claim 4, wherein said analyzer of the domain processor comprises means for extracting digital watermarks from the digital content received from an SU, and means for analyzing the digital watermark to determine if the digital content has been previously marked with the unique identification code of the LCS.

7. (original) The system of claim 4, wherein said system permits the digital content to be stored in the LCS at a degraded quality level if an analysis of the digital content received from the SU concludes that the digital content received from the SU cannot be authenticated because there is no authentication data embedded in the content.

8. (original) The system of claim 4, further comprising at least one SU, each such SU being capable of communicating with the LCS.

9. (original) The system of claim 8, wherein the SU has means to sending a message to the LCS indicating that the SU is requesting a copy of a content data set that is stored on the LCS, said message including information about the identity of the SU, and wherein the LCS comprises:

means to analyze the message from the SU to confirm that the SU is authorized to use the LCS;

means to retrieve a copy of the requested content data set;

means to embed at least one robust open watermark into the copy of the requested content data set, said watermark indicating that the copy is authenticated;

means to embed a second watermark into the copy of the requested content data set, said second watermark being created based upon information transmitted by the SU and information about the LCS; and

means to deliver the watermarked content data set to the SU for its use.

10. (original) The system of claim 8, further comprising a SECD, said SECD capable of receiving a request to transfer at least one data set and capable of transmitting the at least one data set in a secured transmission.

11. (original) The system of claim 10, wherein the SU includes means to send a message to the LCS indicating that the SU is requesting a copy of a content data set that is not stored on the LCS, but which the LCS can obtain from an SECD, said message including information about the identity of the SU;

wherein the SECD comprises:

means to retrieve a copy of the requested content data set;

means to embed at least one robust open watermark into the copy of the requested content data set, said watermark indicating that the copy is authenticated;

means to embed a second watermark into the copy of the requested content data set, said second watermark being created based upon information transmitted by the LCS; and

means to deliver the watermarked content data set to the LCS for its use; and

wherein the LCS comprises:

means to analyze the message from the SU to confirm that the SU is authorized to use the LCS;

means to receive a copy of the requested content data set as transmitted by the SECD;

means to extract at least one watermark to confirm that the content data is authorized for use by the LCS;

means to embed at least one robust open watermark into the copy of the requested content data set, said watermark indicating that the copy is authenticated;

means to embed a second watermark into the copy of the requested content data set, said second watermark being created based upon information transmitted by the SU and information about the LCS; and

means to deliver the watermarked content data set to the SU for its use.

12. (currently amended) The system of claim 8, wherein the SU has means to sending a message to the LCS indicating that the SU is requesting to store a copy of a content data set on a storage unit of the LCS, said message including information about the identity of the SU, and wherein the LCS comprises:

means to analyze the message from the SU to confirm that the SU is authorized to use the LCS;

means <u>to</u> receive a copy of the content data set;

means to determine if a robust open watermark is embedded in the content data set, and to extract the robust open watermark if is it is determined that one exists;

means to analyze any extracted robust open watermarks to determine if the content data set can be authenticated;

means to permit the storage of the content data set on a storage unit of the LCS if i) the LCS authenticates the content data set, or ii) the LCS determines that no robust open watermark is embedded in the content signal.

13. (previously presented) The system of claim 4, further comprising at least one SU, each such SU being capable of communicating with the LCS, and being capable of using only data which has been authorized for use by the SU or which has been determined to be legacy content such that the data contains no additional information to permit authentication.

14. (original) The system of claim 5, wherein the LCS further comprises:

means to embed at least one robust open watermark into a copy of content data, said watermark indicating that the copy is authenticated;

means to embed a second watermark into the copy of content data, said second watermark being created based upon information comprising information uniquely associated with the LCS; and

means to embed a third watermark into the copy of content data, said third watermark being a fragile watermark created based upon information which can enhance the use of the content data on one or more SUs.

15. (original) The system of claim 5, wherein the LCS further comprises:

means for encrypting or scrambling content data, such that content data may be encrypted or scrambled before it is stored in the rewritable storage medium.

16. (previously presented) A system for creating a secure environment for digital content, comprising:

a Secure Electronic Content Distributor (SECD);

a Local Content Server (LCS);

a communications network interconnecting the SECD to the LCS; and

a Satellite Unit (SU) capable of interfacing with the LCS;

said SECD comprising: a storage device for storing a plurality of data sets; an input for receiving a request from the LCS to purchase a selection of at least one of said plurality of data sets; a transaction processor for validating the request to purchase and for processing payment for the request; a security module for encrypting or otherwise securing the selected at least one data set; and an output for transmitting the selected at least one data set that has been encrypted or otherwise secured for transmission over the communications network to the LCS;

said LCS comprising: a domain processor; a first interface for connecting to a communications network; a second interface for communicating with the SU; a memory device for storing a plurality of data sets; and a programmable address module which can be programmed with an identification code uniquely associated with the LCS; and

said SU being a portable module comprising: a memory for accepting secure digital content from a LCS, said digital content comprising data which can be authorized for use or which has been determined to be legacy content such that the data contains no additional information to permit authentication; an interface for communicating with the LCS; and a programmable address module which can be programmed with an identification code uniquely associated with the SU.

17. (previously presented) A method for creating a secure environment for digital content for a consumer, comprising the following steps:

sending a message indicating that a user is requesting a copy of a content data set;

retrieving a copy of the requested content data set;

embedding at least one robust open watermark into the copy of the requested content data set, said watermark indicating that the copy is authenticated;

embedding a second watermark into the copy of the requested content data set, said second watermark being created based upon information transmitted by the requesting user;

transmitting the watermarked content data set to the requesting consumer via an electronic network;

receiving the transmitted watermarked content data set into a Local Content Server (LCS) of the user;

extracting at least one watermark from the transmitted watermarked content data set;

permitting use of the content data set if the LCS determines that use is authorized; and

permitting use of the content data set at a predetermined quality level, said predetermined quality level having been set for legacy content if the LCS determines that use is not authorized.

18. (previously presented) The method of claim 17, wherein the step of permitting use of the content data set if the LCS determines that use is authorized comprises:

checking to see if a watermark extracted from the content data set includes information which matches unique information which is associated with the user; and

permitting the storage of the content data set in a storage unit for the LCS.

19. (previously presented) The method of claim 17, further comprising:

connecting a Satellite Unit (SU) to an LCS,

and wherein the step of permitting use of the content data set if the LCS determines that use is authorized comprises:

checking to see if a watermark extracted from the content data set includes information which matches unique information which is associated with the user; and

embedding a watermark into the content data set using information that is associated with the user and information that is associated with an SU;

delivering the content data set to the SU for its use.

20. (previously presented) A method for creating a secure environment for digital content for a consumer, comprising the following steps:

connecting a Satellite Unit to an local content server (LCS),

sending a message indicating that the SU is requesting a copy of a content data set that is stored on the LCS, said message including information about the identity of the SU;

analyzing the message to confirm that the SU is authorized to use the LCS; and

retrieving a copy of the requested content data set;

assessing whether a secured connection exists between the LCS and the SU;

if a secured connection exists, embedding a watermark into the copy of the requested content data set, said watermark being created based upon information transmitted by the SU and information about the LCS; and

delivering the content data set to the SU for its use, said content data set delivered at a predetermined quality level, said predetermined quality level having been set for legacy content if the LCS determines that use is not authorized.

21. (previously presented) The method of claim 20, further comprising:

embedding an open watermark into the content data to permit enhanced usage of the content data by the user.

22. (previously presented) The method of claim 21, further comprising:

embedding at least one additional watermark into the content data, said at least one additional watermark being based on information about the user, the LCS and an origin of the content data, said watermark serving as a forensic watermark to permit forensic analysis to provide information on the history of the content data's use.

23. (original) The method of claim 20, wherein the content data can be stored at a level of quality which is selected by a user.

24. (previously presented) A method for creating a secure environment for digital content for a consumer, comprising the following steps:

connecting a Satellite Unit (SU) to an local content server (LCS),

sending a message indicating that the SU is requesting a copy of a content data set that is stored on the LCS, said message including information about the identity of the SU;

analyzing the message to confirm that the SU is authorized to use the LCS; and

retrieving a copy of the requested content data set;

assessing whether a secured connection exists between the LCS and the SU;

if a secured connection exists, embedding a watermark into the copy of the requested content data set, said watermark being created based upon information transmitted by the SU and information about the LCS; and

delivering the watermarked content data set to the SU for its use, said watermarked content data set delivered at a predetermined quality level, said predetermined quality level having been set for legacy content if the LCS determines that use is not authorized.

25. (original) The method of claim 24, further comprising:

embedding at least one robust open watermark into the copy of the requested content data set before the requested content data is delivered to the SU, said watermark indicating that the copy is authenticated.

26. (original) The method of claim 25, wherein the robust watermark is embedded using any one of a plurality of embedding algorithms.

27. (original) The method of claim 24, further comprising:

embedding a watermark which includes a hash value from a one-way hash function generated using the content data.

28. (original) The method of claim 25, wherein the robust watermark can be periodically replaced with a new robust watermark generated using a new algorithm with payload that is no greater than that utilized by the old robust watermark.

29. (original) The method of claim 24, further comprising the step of:

embedding additional robust open watermarks into the copy of the requested content data set before the requested content data is delivered to the SU, using a new algorithm; and

re-saving the newly watermarked copy to the LCS.

30. (original) The method of claim 24, further comprising the step of:

saving a copy of the requested content data with the robust watermark to the rewritable media of the LCS.

31. (original) A method for creating a secure environment for digital content for a consumer, comprising the following steps:

connecting a Satellite Unit (SU) to an local content server (LCS),

sending a message indicating that the SU is requesting to store a copy of a content data on the LCS, said message including information about the identity of the SU;

analyzing the message to confirm that the SU is authorized to use the LCS; and

receiving a copy of the content data set;

assessing whether the content data set is authenticated;

if the content data is unauthenticated, denying access to the LCS storage unit; and

if the content data is not capable of authentication, accepting the data at a predetermined quality level, said predetermined quality level having been set for legacy content.

## Information Disclosure Statement

Applicants respectfully submits a copy of EPO Application No. 96919405.9, entitled "Steganographic Method and Device" which corresponds to U.S. Patent 5,613,003 filed June 7, 1995, entitled "Steganographic Method and Device" and a copy of Japanese Patent Application No. 2000-542907 entitled "Multiple Transform Utilization and Application for Secure Digital Watermarking" which corresponds to U.S. Patent Application No. 09/053,628, filed April 2, 1998, entitled "Multiple Transform Utilization and Application for Secure Digital Watermarking".

Applicants thank Examiner Avery for clarification and respectfully request that these references be considered as disclosed in the Information Disclosure Statement ("IDS") dated April 17, 2007.

## REMARKS/ARGUMENTS

The Applicants thank Examiner Avery for the time and consideration to discuss the pending claims and the cited art. These discussions took place on or about June 28, 2007. Examiner Avery acknowledged the differences between the Applicants' claim[s] and Stringer with regards to "legacy content" and "predetermined quality level"—namely, Stringer does not teach how to identify, differentiate or authorize material already possessed by users. Claims 1, 3, 16, 17, 20, 24, and 31 were discussed as having significant advantages over Stringer et al. and the prior art demonstrating patentability over Stringer et al.

## Comments concerning Claim Objections

With regards to Claim 12, Applicants thank Examiner Avery for the helpful comment and have amended Claim 12 to correct the grammatical error.

## Rejections under 35 U.S.C. § 102

### § 102 Rejections based on U.S. Patent 5,341,429 ("Stringer")

Claims 1-31 stand rejected as allegedly anticipated by U.S. Patent No. 5,341,429 issued to Stringer et al. (thereafter "Stringer"). See Page 2 of the final Office Action dated May 9, 2007.

#### Claims 1-31

In order for a reference to anticipate a claim, the reference must disclose each and every feature of the claimed invention, either expressly or inherently, such that a person of ordinary skill in the art could practice the invention without undue experimentation. See *Atlas Powder Co. v. Ireco Inc.*, 190 F.3d 1342, 1347, 51 USPQ2d 1943, 1947 (Fed. Cir. 1999); In *re Paulsen*, 30 F.3d 1475, 1479, 31 USPQ2d 1671, 1673 (Fed. Cir. 1994). Previously Presented Independent Claim 1 recites [emphasis added]: "A **local content server system** (LCS) for creating a secure environment for digital content, comprising: a) a communications port in communication for connecting the system via a network to at least one Secure Electronic Content Distributor (SECD), said SECD capable of storing a plurality of data sets, capable of receiving a request to transfer at least one content data set, and capable of transmitting the at least one content data set in a secured transmission; b) a rewritable storage medium whereby content received from outside the LCS may be stored and retrieved; c) a domain processor that imposes rules and procedures for content being transferred between the LCS and devices outside the LCS; and d) a programmable address module which can be programmed with an identification code uniquely associated with the LCS; and said **domain processor permitting the LCS to receive digital content from outside the LCS provided the LCS first determines that the digital content being delivered to the LCS is authorized for use by the LCS** and if the ***digital content is not authorized for use by the LCS, accepting the digital content at a predetermined quality level, said predetermined quality level having been set for legacy content***." The Section 102 rejection of Claim 1 is improper for at least the reason that Stringer fails to disclose or anticipate (1) "legacy content".

The final Office Action contends that Stringer discloses a conventional local content server ("LCS"), May 9, 2007 final Office Action at Page 2. This contention is respectfully traversed. First, Stringer allegedly teaches a third party that "[t]ransforms the original ephemeral material to its denatured version and wrapper and delivers both to user" (Col. 5 ll. 58-60). Content *received* by users as taught by Stringer, is *identical* to that created by the author. Thus, there is no anticipation that Stringer's alleged LCS could differentiate between users and authors. Specifically, Stringer teaches that a third party "...convert[s] *purchased products to unlimited use and ownership*" (see Stringer at Col. 9 ll. 53-67; Col. 12 ll. 4-12; and Col. 12 ll. 40-48). Thus, the alleged authorization process of Stringer is directed at a transaction *not* determinations concerning admittance of content in a conventional LCS. Second, Stringer fails to disclose any means to differentiate content *already* owned by users— even newly transacted content received by users under Stringer is of "unlimited use and ownership". The claimed invention[s] are directed at handling materials that may lack any identifying information, including legacy content, in a manner consistent with market realities. As taught in the originally filed specification, "it is the user's prerogative to decide how the system will treat non-authenticated content, as well as legacy content". Even, where Stringer allegedly provides identification— it is temporary and controlled by the third party, not the author, and thus may not be reliable. No matter, it is removed. Thus, users can subsequently move content that is identical to the original material. This undermines any alleged utility of Stringer. "To remove the watermark or other material and enable unlimited use of the material, the denatured version of the material is subjected ... to ... any other technique that would serve to erase the watermark from the original material" (Col. 7 ll. 51-57). Applicants respectfully note that the "watermark" of Stringer is not the "digital watermark" of the instant claims, including the various types of watermarks described in the specification and claims, for at the reason that they are *not* removed. Third, by teaching removal of identifying information, Stringer cannot anticipate the LCS of the claims which provides an environment for materials that are essentially identical save the version or status of the data (e.g., *inter alia*, initial, free, legacy, secure, compressed, unsecure, purchased, original, watermarked, signed, hashed, validated, etc.). It logically follows that Stringer fails to anticipate the claim element[s] "receive digital content from outside the LCS provided the LCS first determines that the digital content being delivered to the LCS is authorized for use by the LCS and if the digital content is not authorized for use by the LCS, accepting the digital content at a predetermined quality level". For these additional reasons, Applicants respectfully request the Section 102 rejections be withdrawn.

Additional benefits over Stringer are provided by example and reference to the originally filed specification (*please see for example* Pages 11, 12, 15, 16, 23, 24, 26 & 27 of the originally-filed specification):

> These embodiments may include decisions about availability of a particular good or service through electronic means, such as the Internet, or means that can be modularized ... Consumers may view their anonymous marketplace transactions very differently because of a lack of physical human interactions, but the present invention can enable realistic transactions to occur by maintaining open access and offering strict authentication and verification of the information being traded. This has the effect of allowing legacy relationships, legacy information, and legacy business models to be offered in a manner which more closely reflects many observable transactions in the physical world.

Finally, one of ordinary skill in the art can readily appreciate the widespread existence of content in any number of formats— an example, data released prior to a particular protection scheme or without any use restrictions. Thus, the Applicants additionally traverse the assertion that Stringer or the cited art teaches or anticipates the claim feature: "said predetermined quality level having been set for legacy content". For exemplary purposes, in the case of music, though the present invention[s] are not limited to audio, a "predetermined quality level" (i.e., 44.1 kHz 16 bit) is an example of "legacy content". For purposes of argument, this legacy content is arguably *not* of lesser quality than MP3 or AAC—which were introduced after compact discs and are also compressed. And, Windows 95 may have *arguably* less features than Windows XP. But, Windows 95, being legacy content, is not arguably of lesser quality than Windows XP. The instant invention[s] can handle legacy content and verifiable or secure content seamlessly enabling a more diverse market for information. This is why the Applicants' claims offer significant advantages over Stringer and the cited art.

Because Stringer fails to disclose or anticipate all of the elements of the claims, Claim 1 (and all claims that depend therefrom) is patentable over Stringer and the cited art. For these additional reasons the Section 102 rejections of Claim 1 (and all claims depending therefrom, namely Claim 2) based on Stringer should be withdrawn.

## Additional Comments

### Independent Claim 3 (and all claims depending therefrom, namely Claims 4-15)

Independent Claim 3 includes at least the additional claim element absent in Stringer and the cited art: "said domain processor permitting the LCS to deliver digital content to and receive digital content from an SU that is connected to the LCS's interface, provided the LCS first determines that the digital content being delivered to the SU is authorized for use by the SU or that the digital content being received is authorized for use by the LCS, and if the digital content is not authorized for use, accepting the digital content at a predetermined quality level, said predetermined quality level having been set for legacy content". For the reasons presented with regards to Claim 1 and at least the additional claim elements, Applicants respectfully request the Examiner withdraw the Section 102 rejections for Independent Claim 3 and the claims depending therefrom, namely Claims 4-15.

### Independent Claim 16

Independent Claim 16 includes at least the additional claim element absent in Stringer and the cited art: "said SU being a portable module comprising: a memory for accepting secure digital content from a LCS, said digital content comprising data which can be authorized for use or which has been determined to be legacy content such that the data contains no additional information to permit authentication; an interface for communicating with the LCS; and a programmable address module which can be programmed with an identification code uniquely associated with the SU. For the reasons presented with regards to Claim 1 and at least the additional claim element, Applicants respectfully request the Examiner withdraw the Section 102 rejections for Independent Claim 16.

### Independent Claims 17, 20 & 24 (and all claims pending therefrom, namely Claims 18-19, 21-23, 25-30)

Independent Claim 17 includes at least the additional claim element absent in Stringer and the cited art: "embedding at least one robust open watermark into the copy of the requested content data set, said watermark indicating that the copy is authenticated"; Independent Claim 20 includes at least the additional claim element absent in Stringer and the cited art: "if a secured connection exists, embedding a watermark into the copy of the requested content data set, said watermark being created based upon information transmitted by the SU and information about the LCS"; Independent Claim 24 includes at least the additional claim element absent in Stringer and the cited art: "delivering the watermarked content data set to the SU for its use, said watermarked content data set delivered at a predetermined quality level, said predetermined quality level having been set for legacy content if the LCS determines that use is not authorized".

For the reasons presented with regards to Claim 1, at least the additional claim elements, respectively, and the additional reason that the watermark of Stringer and the cited art is not the watermark of the claims, Applicants respectfully request the Examiner withdraw the Section 102 rejections for Independent Claims 17, 20 & 24 and the claims depending therefrom, namely Claims 18-19, 21-23 & 25-29.

### Independent Claim 31

Independent Claim 31 includes at least the additional claim element absent in Stringer and the cited art: "sending a message indicating that the SU is requesting to store a copy of a content data on the LCS, said message including information about the identity of the SU". For the reasons presented with regards to Claim 1 and at least the additional claim element, Applicants respectfully request the Examiner withdraw the Section 102 rejections for Independent Claim 31.

## Conclusion

Applicants maintain that this application is in condition for allowance, and such disposition is earnestly solicited. Applicants' silence as to the Examiner's comments is not indicative of an acquiescence to the stated grounds of rejection. If the Examiner believes that an interview with the Applicants, either by telephone or in person, would further prosecution of this application, we would welcome the opportunity for such an interview.

It is believed that no other fees are required to ensure entry and consideration of this response.

Respectfully submitted,

Date: July 9, 2007

By: Scott Moskowitz

Scott A. Moskowitz
Tel# (305) 956-9041
Fax# (305) 956-9042

For Blue Spike, Inc.
Scott Moskowitz
Scott A. Moskowitz
President

OIPE
JUL 0 9 2007
IAP87
PATENT & TRADEMARK OFFICE

Your Ref.: 066358.0102JP

Our Ref.: S-1181-1/002365

# JAPANESE TRANSLATION OF PCT APPLICATION

International Patent Application No.

PCT/US99/07262

Date of International Application:

April 2, 1999

## TITLE OF THE INVENTION

Multiple Transform Utilization and Applications
for Secure Digital Watermarking

## INVENTOR

SCOTT A. MOSKOWITZ

## APPLICANT

SCOTT A. MOSKOWITZ

**YUASA AND HARA**

# 受領書

平成１２年１０月　２日
特 許 庁 長 官

識別番号　１０００８９７０５
氏名（名称）　社本　一夫　殿
提出日　平成１２年１０月　２日

以下の書類を受領しました。

| 項番 | 書類名 | 整理番号 | 受付番号 | 出願番号通知（事件の表示） |
|---|---|---|---|---|
| １ | 国内書面 | 002365 | 50001273422 | PCT/US99/ 7262 |

以　上

【書類名】　国内書面

【整理番号】　002365

【提出日】　平成12年10月 2日

【あて先】　特許庁長官殿

【出願の表示】

　【国際出願番号】　PCT/US99/07262

　【出願の区分】　特許

【発明者】

　【住所又は居所】　アメリカ合衆国フロリダ州３３１６０，マイアミ，コリンズ・アベニュー　１６７１１，ナンバー　２５０５

　【氏名】　モスコウィッツ，スコット・エイ

【特許出願人】

　【住所又は居所】　アメリカ合衆国フロリダ州３３１６０，マイアミ，コリンズ・アベニュー　１６７１１，ナンバー　２５０５

　【氏名又は名称】　スコット・エイ・モスコウィッツ

【代理人】

　【識別番号】　100089705

　【住所又は居所】　東京都千代田区大手町二丁目２番１号　新大手町ビル２０６区　ユアサハラ法律特許事務所

　【弁理士】

　【氏名又は名称】　社本　一夫

　【電話番号】　03-3270-6641

【選任した代理人】

　【識別番号】　100071124

　【弁理士】

　【氏名又は名称】　今井　庄亮

【選任した代理人】

　【識別番号】　100076691

　【弁理士】

【氏名又は名称】　増井　忠弐

【選任した代理人】

【識別番号】　100075270

【弁理士】

【氏名又は名称】　小林　泰

【選任した代理人】

【識別番号】　100096013

【弁理士】

【氏名又は名称】　富田　博行

【選任した代理人】

【識別番号】　100087424

【弁理士】

【氏名又は名称】　大塚　就彦

【手数料の表示】

【予納台帳番号】　051806

【納付金額】　21,000円

【提出物件の目録】

【物件名】　明細書の翻訳文　1

【物件名】　図面の翻訳文　1

【物件名】　要約書の翻訳文　1

【プルーフの要否】　要

Proof - 2000/10/02

【書類名】　明細書

【発明の名称】　安全なデジタル透かしのための複数の変換の利用及び適用

【特許請求の範囲】

【請求項１】　メッセージをデジタル情報に符号化する方法であって、前記デジタル情報は複数のデジタル・ブロックを含んでいる、方法において、

前記デジタル・ブロックのそれぞれをスペクトル変換を用いて周波数領域に変換するステップと、

前記変換されたデジタル・ブロックのそれぞれに対して、複数の周波数と関連する振幅とを識別するステップと、

前記デジタル・ブロックのそれぞれに対して、鍵からの基本マスクを用いて、前記識別された振幅の部分集合を選択するステップと、

畳み込みマスクを用いて発生された変換テーブルを用いて、前記メッセージからメッセージ情報を選ぶステップと、

前記選ばれたメッセージ情報に基づいて前記選択された振幅を変更することによって、前記選ばれたメッセージ情報を前記変換されたデジタル・ブロックのそれぞれに符号化するステップと、

を含むことを特徴とする方法。

【請求項２】　請求項１記載の方法において、前記変換するステップは、

高速フーリエ変換を用いて、前記デジタル・ブロックのそれぞれを前記周波数領域に変換するステップを含むことを特徴とする方法。

【請求項３】　請求項２記載の方法において、前記デジタル情報は、画像を形成する複数のカラー・チャネルにおけるピクセルを含み、前記デジタル・ブロックのそれぞれは、前記カラー・チャネルの１つにおけるピクセル領域を表すことを特徴とする方法。

【請求項４】　請求項１記載の方法において、前記デジタル情報はオーディオ情報を含むことを特徴とする方法。

【請求項５】　請求項２記載の方法において、前記識別するステップは、

前記変換されたデジタル・ブロックのそれぞれに対して最大の値を有する所定の数の振幅を識別するステップを含むことを特徴とする方法。

【請求項６】　請求項２記載の方法において、前記選ばれたメッセージ情報はメッセージ・ビットであり、前記符号化するステップは、

前記メッセージ・ビットが真である場合には強度率を用いて前記選択された振幅を減少させ、前記メッセージ・ビットが偽である場合には前記選択された振幅を減少させないことによって、前記選ばれたメッセージ・ビットを前記変換されたデジタル・ブロックのそれぞれに符号化するステップを含むことを特徴とする方法。

【請求項７】　請求項６記載の方法において、前記強度率はユーザによって定義されることを特徴とする方法。

【請求項８】　請求項２記載の方法において、前記選択された振幅と関連する周波数とのそれぞれを前記鍵に記憶するステップを更に含むことを特徴とする方法。

【請求項９】　請求項２記載の方法において、前記デジタル情報の基準部分集合を前記鍵に記憶するステップを更に含むことを特徴とする方法。

【請求項１０】　請求項２記載の方法において、前記デジタル情報は画像を形成するピクセルを含んでおり、更に、

前記ピクセルの基準部分集合を前記鍵にセーブするステップと、

前記画像の元の寸法を前記鍵に記憶するステップと、

を含むことを特徴とする方法。

【請求項１１】　請求項１記載の方法において、前記デジタル情報はオーディオ情報を含んでおり、更に、

オーディオ情報の基準部分集合を前記鍵にセーブするステップと、

前記オーディオ情報の元の寸法を前記鍵に記憶するステップと、

を含むことを特徴とする方法。

【請求項１２】　請求項１０記載の方法において、ピクセルの前記基準部分集合は前記画像におけるピクセルの線を形成することを特徴とする方法。

【請求項１３】　請求項１１記載の方法において、オーディオ情報の前記基準部分集合は振幅設定を含むことを特徴とする方法。

【請求項１４】　請求項８記載の方法において、前記画像は矩形であり、ピ

クセルの前記基準部分集合は前記矩形の対角線を形成することを特徴とする方法。

【請求項１５】　請求項２記載の方法において、
所定の鍵が前記符号化されたメッセージ情報を復号化することを要求するステップを更に含むことを特徴とする方法。

【請求項１６】　請求項２記載の方法において、
公開鍵の対が前記符号化されたメッセージ情報を復号化することを要求するステップを更に含むことを特徴とする方法。

【請求項１７】　請求項２記載の方法において、
前記メッセージに対する元のハッシュ値を計算するステップと、
前記元のハッシュ値を前記鍵に記憶するステップと、
を更に含むことを特徴とする方法。

【請求項１８】　鍵を用いてでる情報をデスケーリングする方法であって、
前記デジタル情報の元の寸法を前記鍵から決定するステップと、
前記デジタル情報を前記元の寸法にスケーリングするステップと、
情報の基準部分集合を前記鍵から取得するステップと、
前記基準部分集合を前記スケーリングされたデジタル情報における対応する情報と比較するステップと、
を含むことを特徴とする方法。

【請求項１９】　請求項１８記載の方法において、デスケーリングされる前記デジタル情報はデジタル画像であり、前記鍵から情報の基準部分集合を取得するステップは前記鍵からピクセルの基準部分集合を取得するステップを含むことを特徴とする方法。

【請求項２０】　請求項１８記載の方法において、デスケーリングされる前記デジタル情報はオーディオ・デジタル情報であり、前記鍵から情報の基準部分集合を取得するステップは前記鍵からオーディオ情報の基準部分集合を取得するステップを含むことを特徴とする方法。

【請求項２１】　請求項１９記載の方法において、前記比較するステップは前記比較に基づいて第１の適合する値を決定し、この方法は、更に、

前記スケーリングされたデジタル画像をパッド・ピクセルのエリアを用いてパディングするステップと、

ピクセルの前記基準部分集合を前記パディングされた画像における対応するピクセルと再度比較して第２の適合する値を決定するステップと、

を含むことを特徴とする方法。

【請求項２２】　請求項２０記載の方法において、パッド・ピクセルの前記エリアは、単一のピクセルのローであることを特徴とする方法。

【請求項２３】　請求項２０記載の方法において、パッド・ピクセルの前記エリアは、単一のピクセルのコラムであることを特徴とする方法。

【請求項２４】　請求項２０記載の方法において、前記パディング及び再度比較するステップは複数回実行されることを特徴とする方法。

【請求項２５】　請求項２０記載の方法において、前記決定された適合する値の中で最良の適合する値を選び、前記デジタル画像を元のサイズに回復し、前記最良の適合する値と関連する任意のパッド・ピクセルを含むステップを更に含むことを特徴とする方法。

【請求項２６】　所定の鍵を用いて符号化されたデジタル情報からメッセージを抽出する方法であって、

前記所定の鍵を用いて、前記符号化されたデジタル情報を複数のデジタル・ブロックを含むデジタル情報に復号化するステップと、

スペクトル変換を用いて、前記デジタル・ブロックのそれぞれを周波数領域に変換するステップと、

前記変換されたデジタル・ブロックのそれぞれに対して、複数の周波数と関連する振幅とを識別するステップと、

前記鍵からの基本マスクを用いて、前記変換されたデジタル・ブロックのそれぞれに対して、前記識別された振幅の部分集合を選択するステップと、

前記選択された振幅と前記所定の鍵に記憶された元の振幅とを比較し、符号化されたメッセージ情報の位置を決定するステップと、

前記符号化されたメッセージ情報と逆変換テーブルとを用いて、前記メッセージをアセンブルするステップと、

を含むことを特徴とする方法。

【請求項２７】　請求項２６記載の方法において、前記変換するステップは

、

高速フーリエ変換を用いて、前記デジタル・ブロックのそれぞれを周波数領域に変換するステップを含むことを特徴とする方法。

【請求項２８】　請求項２７記載の方法において、

前記アセンブルされたメッセージに対するハッシュ値を計算するステップと、

前記計算されたハッシュ値を前記所定の鍵の中の元のハッシュ値と比較するステップと、

を更に含むことを特徴とする方法。

【請求項２９】　鍵を用いてデジタル信号をデスケーリングする方法であって、

前記鍵から前記デジタル信号の元の寸法を決定するステップと、

前記デジタル信号を前記元の寸法にスケーリングするステップと、

前記鍵から基準信号部分を取得するステップと、

前記基準信号部分を前記スケーリングされた信号における対応する信号部分と比較するステップと、

を含むことを特徴とする方法。

【請求項３０】　デジタル信号を保護する方法であって、

伝達関数ベースのマスク・セットと元のデジタル信号のオフセット座標値とから構成される所定の鍵を作成するステップと、

前記デジタル信号を前記所定の鍵を用いて符号化するステップと、

を含むことを特徴とする方法。

【請求項３１】　請求項３０記載の方法において、前記デジタル信号は連続的なアナログ波形を表すことを特徴とする方法。

【請求項３２】　請求項３０記載の方法において、前記所定の鍵は複数のマスク・セットを含むことを特徴とする方法。

【請求項３３】　請求項３０記載の方法において、前記マスク・セットは、公開鍵と秘密鍵とを含む鍵の対によって暗号化されることを特徴とする方法。

【請求項３４】　請求項３０記載の方法において、

デジタル透かし技術を用いて前記デジタル信号に関する権利者、使用又はそれ以外の情報を識別する情報を前記デジタル信号の中に符号化するステップを更に含むことを特徴とする方法。

【請求項３５】　請求項３０記載の方法において、前記デジタル信号は静止画像、オーディオ又はビデオを表すことを特徴とする方法。

【請求項３６】　請求項３０記載の方法において、

ランダム又は疑似ランダムな一連のビットを有する１つ又は複数のマスクを含むマスク・セットを選択するステップと、

前記マスク・セットを、前記伝達関数ベースのマスク・セットの開始において有効化するステップと、

を更に含むことを特徴とする方法。

【請求項３７】　請求項３６記載の方法において、前記有効化するステップは、

前記伝達関数ベースのマスク・セットの開始において計算されたハッシュ値を前記ハッシュ値の所定の伝達関数と比較するステップを含むことを特徴とする方法。

【請求項３８】　請求項３６記載の方法において、前記有効化するステップは、

前記伝達関数ベースのマスク・セットの開始におけるデジタル署名を前記デジタル署名の所定の伝達関数と比較するステップを含むことを特徴とする方法。

【請求項３９】　請求項３６記載の方法において、

デジタル透かし技術を用いて前記デジタル信号に関する権利者、使用又はそれ以外の情報を識別する情報を前記デジタル信号の中に埋め込むステップを更に含み、

前記有効化するステップは、前記埋め込まれた情報の有効化に依存することを特徴とする方法。

【請求項４０】　請求項３０記載の方法において、

前記デジタル信号においてキャリア信号データの安全な一方向ハッシュ関数を

計算するステップを更に含んでおり、前記ハッシュ関数は、前記伝達関数ベースのマスク・セットを搬送する目的で前記キャリア信号の中に導入された変化を感知しないことを特徴とする方法。

【請求項４１】　デジタル信号を保護する方法であって、

伝達関数ベースのマスク・セットと元のデジタル信号のオフセット座標値とで構成された所定の鍵を作成するステップと、

正しい伝達関数ベースのマスク・セットを含む前記所定の鍵を前記データの再生の間に認証するステップと、

前記データの再生を測定してコンテンツをモニタし、前記デジタル信号が変更されたかどうかを判断するステップと、

を含むことを特徴とする方法。

【請求項４２】　請求項３０記載の方法において、前記デジタル信号はビット・ストリームであり、この方法は、更に、

符号化のために用いられ、ランダム基本マスクと、ランダム畳み込みマスクと、メッセージ・デリミタのランダム開始とを含む複数のマスクを発生するステップと、

符号化されるメッセージ・ビット・ストリームを発生するステップと、

前記メッセージ・ビット・ストリームと、ステガ・サイファ・マップ真理テーブルと、前記基本マスクと、前記畳み込みマスクと、メッセージ・デリミタの前記開始とをメモリにロードするステップと、

基本マスク・インデクスと、畳み込みマスク・インデクスと、メッセージ・ビット・インデクスとの状態を初期化するステップと、

前記メッセージ・ビット・ストリームにおける全ビット数と等しくなるようにメッセージ・サイズを設定するステップと、

を含むことを特徴とする方法。

【請求項４３】　請求項４２記載の方法において、前記デジタル情報は複数のウィンドウを有しており、この方法は、更に、

サンプル・ストリームにおけるどのウィンドウの上で前記メッセージが符号化されるかを計算するステップと、

前記計算されたウィンドウにおける情報の安全な一方向ハッシュ関数を計算するステップであって、前記ハッシュ関数はステガ・サイファによって導かれるサンプルにおける変化を感知しないハッシュ値を発生する、ステップと、

データの符号化されたストリームにおける前記計算されたハッシュ値を符号化するステップと、

を含むことを特徴とする方法。

【請求項４４】　請求項４０記載の方法において、前記選択するステップは、

ランダム・タイピングにおけるキーボード・レイテンシ期間から導かれた一連のランダム・ビットを収集するステップと、

初期の一連のランダム・ビットをＭＤ５アルゴリズムを介して処理するステップと、

前記ＭＤ処理の結果を用いて、トリプルＤＥＳ暗号化ループを供給し、各サイクルの後のそれぞれの結果の最下位ビットを抽出するステップと、

前記トリプルＤＥＳ出力ビットをランダムな一連のビットの中に連結するステップと、

を含むことを特徴とする方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、デジタル情報の保護に関する。更に詳しくは、本発明は、安全なデジタル透かしのための複数の変換の利用及び適用に関する。

【０００２】

【関連出願への相互参照】

本発明は、１９９６年１月１７日に出願された米国特許出願第０８／５８７，９４３号"Method for Stega-Cipher Protection of Computer Code"に基づいて優先権を主張している。この米国特許出願の開示のすべてを、本出願において援用する。

【０００３】

【従来の技術】

商業的に価値のある情報が「デジタル」形式で制作され記憶されることが増加している。例えば、音楽、写真及び画像のすべてが、１及び０などの一連の数として記憶され伝送されることが可能である。デジタル技術によると、元の情報を非常に正確に再生することができる。しかし、不運なことに、デジタル技術によると、その持ち主の許可を得ることなく、情報を容易にコピーすることもできるのである。

【０００４】

デジタル透かし（電子透かし、digital watermark）は、デジタル化されたマルチメディア・コンテンツの制作者（creators）と出版業者（publishers）とがコンテンツのローカルで安全な識別及び認証を要求する収束点に存在している。侵害行為（piracy）は貴重なデジタル情報の流通を損なう方向に作用するから、そのような作品のコピーや二次的（derivative）なコピーに対する責任を確立することが重要である。デジタル透かしシステムの目的は、基礎となるコンテンツ信号の中に、ほとんど又は全く痕跡を残すことなく、そして知覚可能であることが標準となるように、与えられた１つ又は複数の情報信号を挿入することである。その際に、基礎となる信号における符号化レベルと位置感度（location sensitivity）とを最大化することにより、この透かしを除去しようと試みるとコンテンツ信号に強制的に損傷が生じるようになっている。「マスタ」、ステレオ、ＮＴＳＣ（National Television Standards Committee）ビデオ、オーディオ・テープ又はコンパクト・ディスクであるかどうかなど、マルチメディア・コンテンツの様々な形態を考慮すると、質に関する寛容度は、個人ごとに変動し、そのコンテンツの基礎となる商業的及び美的な価値に影響を与える。従って、著作権、所有権（ownership right）、購入者情報又はこれらの何らかの組合せや関連データをそのコンテンツの中に結合させ、それにより、それが商業的であってもそれ以外の態様であっても認証されていない流通がそれ以後なされる場合には、そのコンテンツが損傷を受け、従って、その価値が低下するようにすることが望ましい。デジタル透かしは、このような関心の多くに向けられたものであり、この技術分野における研究は、これまでに、極めて堅固で安全な実現に対する豊かな

基礎を提供してきている。

【０００５】

特に関心が向けられているのは、コンテンツのデジタル化された「作品」（piece）の価値とそのコンテンツに値する「保護」を提供するためのコストとのバランスである。現実の世界における経済行動と並行するように、商業銀行の安全性（セキュリティ）を知覚できるからといって、銀行預金をするのに要する費用及び時間のために、人々は直ちに現金を銀行に預金するということにはならない。ほとんどの個人にとっては、１００米ドルをもっているからといって、それを財布にしまっておく以上の保護が必要とされることはない。また、ワールド・ワイド・ウェブ（WWW）すなわちウェブが存在するからといって、オーディオや、静止画像等の媒体のようなデジタル化することができる媒体に対して価値が創造されたことを意味しない。ウェブは、単に、情報交換のための媒体であり、コンテンツの商業的な価値を決定することはない。しかし、媒体を交換するためにウェブを用いることにより、その価値を決定するのに役立つ情報が提供されるため、デジタル化されたコンテンツに対する責任が要求される。デジタル透かしは、このプロセスにおけるツール（道具）であって、著作権などの法的権利に関するより公的な課題を確立するそれ以外の機構に代わるものではないことに注意してほしい。例えば、デジタル透かしは、コンテンツの価値を判断する際の「履歴平均」（historical average）アプローチに代わるものではない。これは、コンテンツの知覚された価値だけに基づいて購入をしようとする個人の市場（マーケット）のことである。例えば、インターネット又はそれ以外の任意の電子的な交換手段を介して写真が流通しても、その写真の基礎的な価値が増加することは必ずしもない。しかし、そのような形式の「放送」によってより大きな観客に到達する機会が生じることは、「潜在的」により大きな市場に基づく価値を生じさせる望ましい機構でありうる。この決定は、当該権利者のみが唯一なすことができる。

【０００６】

実際、多くの場合に、コンテンツの時間的な価値に依存して、アクセスが適切に制御されていない場合には、価値が現実に低下することがありうる。月刊誌と

して販売されている雑誌の場合には、その雑誌が販売されている期間を超えて、その雑誌に掲載されている写真の価値を評価することは困難である。コンパクト・ディスクの価値に関しても、同様な時間に関する変動要素があるし、デジタル化されたオーディオ信号のパッケージングとパッケージを伴わない電子的な交換とのような有形的な変動要素もある。インターネットは、単に、消費者により迅速に到達する手段を提供するだけであって，それ以外の「市場に基づく」価値に取って代わるものではない。デジタル透かしは、適切に実現されるのであれば、権利者の決定に関する必要な層を追加することになり，デジタル透かしが「証明可能な程度に安全」（provably secure）であるときには、価値を決定し評価する際に大いに役立つ。本発明は、デジタル透かし技術の改良であり、現実世界における商品の真偽判定方法と類似する態様で、デジタル化されたコンテンツを「改ざん不能」（tamper-proof）にする手段を与える。

【０００７】

デジタル透かし技術における一般的な弱点は、透かしを実現する方法に関する。ほとんどのアプローチにおいて、保護されるべき作品の制作者ではなくデジタル透かしを実現する者に、検出及び復号制御に関して依存している。様々な透かし技術が有するこの基本的側面のために、第三者がそのようなデジタル透かしの実現を成功裏に利用する際には、この技術の改良に対する適切な経済的インセンティブが失われる。特定の形式の利用がいったんなされると、それ以後の透かしの検出が曖昧になる。そして、それ以後の時点において同じ透かしプロセスを用いた符号化を成功であると見なすことになる。

【０００８】

安全なデジタル透かしのいくつかの実現例がこの基本的な制御の課題に取り組んでおり、「キー・ベース」（key-based）のアプローチの基礎を形成している。これらは、以下の米国特許及び出願中の米国特許出願がカバーしている。すなわち，"Steganographic Method and Device"と題する米国特許第５，６１３，００４号及びそれから生じた米国特許出願第０８／７７５，２１６号；"Human Assisted Random Key Generation and Application for Digital Watermark System"と題する米国特許出願第０８／５８７，９４４号；"Method for Stega-Cipher

Protection of Computer Code"と題する米国特許出願第０８／５８７，９４３号；"Optimization Methods for the Insertion, Protection, and Detection of Digital Watermarks in Digital Data"と題する米国特許出願第０８／６７７，４３５号；及び"Z-Transform Implementation of Digital Watermarks"と題する米国特許出願第０８／７７２、２２２号である。これらの米国特許及び米国特許出願における開示内容は本出願において援用する。公開鍵暗号システムは、米国特許第4，２００，７７０号、第4，２１８、５８２号、第4，４０５，８２９号及び第4，４２４，４１４号に記載されている。これらの米国特許における開示内容は、本出願において援用する。

【０００９】

これらのデジタル透かしによるセキュリティ方法を改良することによって、複数の変換を用い、信号特性を操作し、必要な関係を符号化及び復号化動作に用いられるマスク・セットすなわち「鍵」に適用することが、これらの方法の最適化された組合せとして考察される。透かしの符号化は、符号化アルゴリズムにおいて用いられる変換に関して最終的にほんの僅かに異なるが、公開された分散型のアーキテクチャというより大きな課題によって、抹消しようとする試みに打ち勝つ、より堅固なアプローチが要求され、更には、透かしの検出を不可能にする手段が要求される。これらの「攻撃」は、計算論的に比較すると、正反対な態様（diametrically）で関連している。例えば、クロッピング（cropping）とスケーリング（scaling）とは、信号処理の向きが異なり、結果的には特定の透かしアプローチを脆弱化する可能性があるが、すべての透かしアプローチについてはそういうことはない。

【００１０】

ブロック・ベース又は全体のデータ・セット変換のいずれかを用いて符号化を行う現時点で利用できるアプローチは、必ず、空間領域又は周波数領域のどちらか一方においてデータを符号化するが、両方の領域においてそうすることは決してない。同時的なクロッピング及びスケーリングは、空間及び周波数領域に影響し、それによって、使用可能な透かしシステムのほとんどを曖昧にする。複数の操作を生き延びる能力は、透かしの入れられた媒体のセキュリティを確実にしよ

うとしている者にとっては明確な利点である。本発明は、鍵ベースのアプローチを用いて既存の透かしを改良することを目指している。その際に、それ以後に透かしが入れられるコンテンツを権利者やコンテンツ制作者がより広く制御できるようにする。

【００１１】

現時点で利用可能な多くの静止画透かしアプリケーションは、鍵ベースの実現例とは根本的に異なっている。これらの製品としては、デジマーク（Digimarc）社やシグナム（Signum）社による製品があるが、これらの製品は、復号化動作に関してはオリジナルの画像との比較に完全に依存している透かしメッセージを符号化することによって、堅固（robust）な透かしを提供することを目指している。ブロックごとに実行される離散コサイン変換である変換のそれ以後の結果は、デジタル的に符号が付される。埋め込まれた透かしは、画像の知覚的な質とは全く関係がなく、従って、一般的に利用可能なデコーダの逆方向の適用が、攻撃の非常によい最初のラインとなる。同様にして、符号化プロセスは、第三者によって適用されることもありうる。これは、いくつかの堅固性のテストにおいて示されているように、或るプロセスを用いて他のプロセスを用いて透かしが入れられた画像の結果を符号化するものである。透かしを放棄しないこと（nonrepudiation）はできない。その理由は、デジマーク社とシグナム社とが、画像の権利に関するすべての登録の機関として機能しているからである。

【００１２】

攻撃の別のラインとして、エラーのない検出が困難又は不可能であるように追加されている高周波ノイズの一部を除去するローパス・フィルタがある。最終的には、単純なＪＰＥＧ変換の多くのテストがこのような透かしは生き延びることができないことを示す。その理由は、ＪＰＥＧが、透かしを入れるプロセスによって用いられる符号化変換と同じ変換に基づいているからである。これ以外の注意すべき実現例としては、例えば、ＮＥＣの研究者たちによって開発されたシグナファイ（Signafy）によるものなどがあるが、画像の全体の変換を実行することによって、透かしメッセージを符号化しているようである。このプロセスの目的は、画像の「候補となる」透かしビット又は領域をより一貫性をもって識別し

て、信号の知覚的に著しい領域において符号化を行うことである。そうであっても、シグナファイは、復号化を達成するのに、オリジナルの透かしの入れられていない画像に依存する。

【００１３】

これらの方法は、すべてが、透かしを比較的エラーのない態様で検出することを確実にするために、オリジナルの透かしの入れられていない画像に依然として依存している。ステガノグラフィック（steganographic）な方法では、復号化動作のためにその媒体のオリジナルな透かしの入れられていないコピーを用いることなく透かしのセキュリティを提供すると共に、ユーザに暗号化された鍵を用いて暗号的なセキュリティをも提供することが目的とされる。すなわち、符号化動作と復号化動作とのために、同じ鍵が用いられる。それぞれのユーザが非対称的な符号化及び復号化動作を実行するための公開／秘密鍵対を有するような公開鍵対を用いることもできる。公開鍵暗号に関する議論と暗号化に関する利点とは、広く文書化がなされている。公開鍵インフラストラクチャの利用可能性が増加していることは、証明可能なセキュリティを認識しうるということを示している。透かしの実現化がこのように鍵ベースであることにより、セキュリティについては鍵に依存することが可能であり、それによって、透かしメッセージと透かしの入れられたコンテンツとのセキュリティ及び認証に対する多層化（layered）されたアプローチが得られる。

【００１４】

これ以外の実現例が生き延びること（survivability）に対するに対する攻撃も容易に利用可能であることが知られている。透かしメッセージに対する興味深いネットワーク・ベースの攻撃も知られているが、これは、中央の登録サーバを騙して、画像が登録されている権利者とは別の誰かが権利を有していると想定させるものである。また、これによると、集中的な透かし技術は十分に堅固なものではなく、マルチメディア作品のデジタル化されたコピーの権利者に関する適切な確認を行うことはできないという懸念が現実のものとなる。

【００１５】

【発明が解決しようとする課題】

複数の変換を実行することに関する計算論的な要求は、静止画やオーディオなどのある種の媒体にとっては禁止されないのであるから、本発明は、復号化を実行するのにオリジナルの透かしの入れられていないコピーを必要とすることなしに、媒体に確実に透かしを入れる手段を提供することを目的とする。これらの変換は、コンテンツの観察者又は権利者に対して単純には明らかでない態様で実行することができる。しかし、これらの観察者や権利者は、透かしが依然として検出可能であると考えることができる。更に、特定の媒体のタイプが一般的に圧縮されている場合（ＪＰＥＧ、ＭＰＥＧなど）には、複数の変換を用いて、透かしを入れるプロセスに先立ってマスク・セットを適切に設定し、透かしの入れられた従って知覚された「安全」なコピーを未知の第三者に解放する前に、ユーザに生き残り可能性について警告することができる。本発明の結果は、透かしへのより現実的なアプローチであって、鍵の証明可能なセキュリティだけでなく媒体のタイプも考慮している。従って、電子商取引のためのより信頼性の高いモデルも可能である。

【００１６】

透かしを挿入するために最適化された「封筒」を作成し、デジタル的にサンプリングされたコンテンツに対する確実な責任を確立することにより、大きな透かしセキュリティの基礎が得られるが、これは、本発明の補助的な目的である。発生される所定の又はランダムな鍵は、隠された情報信号にアクセスするために不可欠な地図であるだけではなく、オリジナルな信号の部分集合であって、それにより、オリジナルな信号との比較が不要になる。これによって、デジタル透かしの全体的なセキュリティが向上する。

【００１７】

同時的なクロッピング及びスケーリングが生き延びること（生き残ること、survival）は、画像及びオーディオ透かしに関しては、困難である。というのは、そのような変換は、画像やオーディオの偶然的（inadvertent）な使用と、透かしへの意図的な攻撃とで共通だからである。対応の効果は、オーディオの場合にはるかに明らかであるが、広帯域の変動などのように狭い意味で「周波数ベース」である透かしは、作品の元の長さから「クロッピング」又はクリップされたオ

ーディオ・サンプルにおけるアライメントの問題を有している。スケーリングは、人間の聴覚系にとってはるかにより顕著であるが、僅かな変化が、消費者には明らかではないにもかかわらず、周波数だけのタイプの透かしに影響することがありうる。ほとんどが周波数ベースの埋め込み形信号処理である、利用可能なオーディオ透かしアプリケーションに対するはるかに大きな脅威は、時間ベースの変換であり、これには、オーディオ信号の時間ベースの圧縮及び解凍が含まれる。シグナファイは、広帯域ベースの透かしの例であり、ソラナ（Solana）テクノロジ、ＣＲＬ、ＢＢＮ、ＭＩＴなどによるアプリケーションも同様である。「空間領域」アプローチというのが、デジマルク、シグナム、ＡＲＩＳ、アービトロン（Arbitron）などによって開発された技術に対するより適切な名称である。興味深いことに、時間ベースのアプローチは、画像について考察される場合には、基本的には空間ベースのアプローチである。ピクセルは、「畳み込み的」（convolutional）である。これら間の差異は、周波数の広帯域化された（spread-spectrum-ed）領域は「あまりに」うまく定義されているために、埋め込まれた信号と同じサブバンドでのランダム・ノイズの過剰な符号化を受けることになるという点である。

【００１８】

ジョバンニ（Giovannni）は、現実の透かしに対して、ブロック・ベースのアプローチを用いる。しかし、それには、スケーリングされた画像をその元のスケールに回復させることができる画像認識が伴っている。この「デスケーリング」は、画像が復号化される前に適用される。他のシステムでは、元の画像を透かし入りの画像と「区別」して「デスケーリング」を行っている。デスケーリングが、あらゆる画像、オーディオ又はビデオ透かしの生き残りにとって固有の重要性を有していることは明らかである。明らかでないのは、区別の動作がセキュリティの見地から受け入れ可能であるか、ということである。更に、画像のユーザ又は制作者ではなく、透かし「機関」によって区別が実行されなければならない場合には、権利者は、元の透かしの入っていないコンテンツを支配できないことになる。符号化／復号化鍵／鍵の対の内部でマスク・セットを用いることとは別に、元の信号を用いなければならない。オリジナルは、検出及び復号化を実行する

のに必要であるが、以上で説明した攻撃に関しては、透かしの入れられたコンテンツに対する権利を明確に確立することは不可能である。

【００１９】

以上を鑑みると、以上で論じた課題を解決する安全なデジタル透かしのための複数の変換の利用及び適用に対する実質的な必要性が存在することを理解することができるであろう。

【００２０】

【課題を解決するための手段】

安全なデジタル透かしのための複数の変換の利用及び適用によってこの技術における短所は大幅に改善することができる。本発明の或る実施例では、保護されるべきデジタル情報におけるデジタル・ブロックは、高速フーリエ変換を用いて周波数領域に変換される。複数の周波数及び関連する振幅が、変換されたデジタル・ブロックのそれぞれに対して識別され、識別された振幅の部分集合が、鍵からの基本マスクを用いてデジタル・ブロックのそれぞれに対して選択される。メッセージ情報は、畳み込みマスクを用いて発生された変換テーブルを用いて、メッセージから選択される。選ばれたメッセージ情報は、選択されたメッセージ情報に基づいて選択される振幅を変化させることによって、変換されたデジタル・ブロックのそれぞれに符号化される。

【００２１】

以下で明らかになる本発明のこれらの及びそれ以外の効果及び特徴により、本発明の性質は、以下で行う本発明の詳細な説明と、冒頭の特許請求の範囲と、添付の図面とを参照することによって、より明確に理解することができるはずである。

【００２２】

【発明の実施の形態】

本発明の或る実施例によると、安全なデジタル透かしのために複数の変換が用いられる。周波数領域又は空間領域の変換を用いる透かしには２つのアプローチが存在する。すなわち、小さなブロックを用いる場合とデータ・セット全体を用いる場合とである。オーディオやビデオのような時間ベースの媒体に対しては、

小さな部分において作業するのが実際的である。というのは、ファイル全体では、サイズが数メガバイトにもなりうるからである。しかし、静止画については、ファイルははるかに小さいのが通常であり、１回の操作で変換することができる。２つのアプローチは、それぞれが、各自の利点を有している。ブロック・ベースの方法は、クロッピングに対する抵抗性を有する。クロッピング（cropping）というのは、信号の部分的な切り取り又は除去である。データは複数の小さな部分（piece）に記憶されるので、クロッピングは、単に、いくつかの部分が失われることを意味する。１つの完全な透かしを復号化するのに十分なブロックが残っている限り、クロッピングによって、その透かしが除去されることはない。しかし、ブロック・ベースのシステムは、スケーリングに弱い。アフィン・スケーリング（affine scaling）又は「収縮」（shrinking）などのスケーリングは、信号の高周波の損失につながる。ブロックのサイズが３２サンプルであり、データが２００％スケーリングされる場合には、関係のあるデータは、６４サンプルをカバーすることになる。しかし、デコーダは、依然として、データは３２サンプルにあると考えるので、透かしを適切に読み取るのに必要な空間の半分しか用いない。セット全体のアプローチは、逆の振る舞いを有する。このアプローチは、スケーリングを生き延びるのは非常に得意である。その理由は、このアプローチでは、データを全体として扱い、符号化の前にデータを特定のサイズにスケーリングするのが一般的であるからである。しかし、どのように小さなクロッピングであっても、変換のアライメントを混乱させ、透かしを曖昧にしてしまう可能性がある。

【００２３】

本発明を用いると、そして、これまでに開示されている材料を組み入れることによって、符号化鍵／鍵の対を用いて画像や歌やビデオを認証し、暗号による誤った肯定的な一致を排除し、オリジナルな透かしの入れられていない作品の代わりに第三者の権限を備えた登録を通じて著作権の通信を提供することが可能となる。

【００２４】

本発明は、従来技術に対する明らかな改良を提供するのであるが、元（オリジ

ナル）の信号の座標値を鍵の上にオフセットし、次にそれを用いてユーザ又は認証を受けた「鍵の持ち主」による復号化又は検出動作が行われることによって、過去に開示された内容に対する改良がなされる。このオフセットは、透かしが、成功裏に符号化されうるデータの量を、シャノンのノイズを含むチャネルの符号化定理に基づいて「運ばせる」（ペイロードさせる）ことができるコンテンツにおいて必要であり，これによって、透かしメッセージを有する信号の十分に不可視的な「飽和」が回避され，権利者が単一のメッセージを検出することが可能となる。例えば、或る画像が単一の１００ビットのメッセージ又は１２のＡＳＣＩＩ文字を運ぶのに十分なペイロードだけを有するというのも、全くありうることである。本発明の発明者によってテストがなされたオーディオでの実現例では、毎秒１０００ビットが、１６ビットの４４．１ｋＨｚのオーディオ信号において、不可聴的に符号化される。電子的に利用可能なほとんどの画像は、同じ「ペイロード」率を与えることができるほどに十分なデータを有していない。従って、クロッピング及びスケーリングが同時に生き延びることは画像の場合の方が、それに対応する商業的に利用可能なオーディオ又はビデオ・トラックの場合よりも困難であることになる。追加されるセキュリティの効果は、広帯域又は周波数のみのアプリケーションに基づく透かしシステムのランダマイザが制限されているほど、透かしデータのランダム値は、制限された信号帯域上で「ホッピング」することになり、また、鍵もまた，ランダムな態様でより効果的に符号化を行うのに用いられる暗号化された又はランダムなデータの独立なソースである、ということである。鍵は、実際に，ビット数で測定した場合に、透かしメッセージ自体よりも大きなランダム値を有しうる。透かしデコーダは、画像が、そのオリジナルのスケールに含まれていることを求められ，また、その「デスケーリング」された寸法に基づいてクロッピングされたかどうかを決定することができる。

【００２５】

コンテンツに透かしを入れそのコンテンツの流通を有効化するために鍵を要求するシステムの利点は明らかである。異なる情報を符号化するには異なる鍵を用いることができる。その際に，安全な一方向ハッシュ関数や、デジタル署名や，更には一時的パッド（one-time pads）でさえも鍵の中に組み入れることによっ

て、埋め込まれた信号を保護し、透かしの入れられた画像とその鍵／鍵の対を拒絶せずに有効化することができる。後に、これらの同じ鍵を用いて、埋め込まれたデジタル署名だけを後で有効化する、又は、デジタル透かしメッセージを完全に復号化する。コンテンツにデジタル透かしが入れられているということだけでなく、流通業者はそれ以外にはどのような機能も有していない鍵を用いてデジタル署名のチェックを実行することによって透かしの有効性をチェックしなければならないということも、出版業者は、容易に要求することができる。

【００２６】

安全なデジタル透かしが、いくらか論じられ始めている。レイトン（Leighton）は、米国特許第５，６６４，０１８号に、デジタル透かしにおける共謀的な攻撃（collusion attack）を防止する手段を記載している。しかし、レイトンは、記載されているセキュリティを現実的には提供できない可能性がある。例えば、透かし技術が線形であるような特定の場合には、「挿入封筒」又は「透かし空間」が矛盾なく定義されており（well-defined）、従って、認証を受けていないものによる共謀よりは複雑でない攻撃を受ける可能性がある。透かし符号化レベルにおける過剰符号化（over encoding）は、そのような線形の実現例における１つの単純な攻撃に過ぎない。レイトンによって無視された別の考慮として、商業的価値のあるコンテンツは、多くの場合に、既に透かしの入れられていない形態でいずれかの場所に既に存在しており、潜在的な侵害行為に容易にさらされる状態にあるので、どのようなタイプの共謀行為も不要であるということがある。この例として、コンパクト・ディスクやデジタル放送されたビデオなど多くがある。透かしデータの前処理を用いて埋め込まれた信号にデジタル署名をすることによって、共謀の成功を回避することができる可能性が大きい。透かしを入れる媒体に依存するが、非常に個別化された（granular）透かしアルゴリズムは、ベースラインとなる透かしが何らかの機能を有しているという予測よりも、デジタル的にサンプリングがなされるあらゆる媒体において共通な与えられた量子化人工物を、何か観測可能なものよりも低いレベルで成功裏に符号化できる可能性が高い。

【００２７】

更に、ここで開示されている「ベースライン」透かしは、かなり主観的なものである。これは、この技術分野のいずれかの場所で信号の「知覚的に意義のある」領域として説明されるだけである。すなわち，透かし関数の線形性を減少させる，又は，透かしの挿入を反転させることにより、「ベースライン」透かしを阿ｓくせいするのに要求される追加的な作業なしに同じ効果が得られるように思われる。実際、透かしアルゴリズムは、追加的なステップなしに、ターゲット挿入封筒又は領域を既に定義することができるべきである。更に、本発明の発明者によって既に開示されている出願では、透かしデータに加えて、利用可能な透かし領域の「ビット空間」又は符号化とは関係のないランダム・ノイズよりも少ないビットを符号化するように設定することにより、可能性のある攻撃やそれ以外の抹消の試みを混乱させることができる透かし技術が説明されている。「候補ビット」の領域は、任意の数の圧縮方式又は変換によって定義することができ、すべてのビットを符号化するすることは必要でない。更に、すべてのビットを符号化することは、符号化方式を知りながら領域を複製することができるものにとっては、現実的には、セキュリティ上の弱点として作用する可能性がある。やはり、セキュリティは、実際の透かしメッセージの外部にオフセットされていなければならず、それによって、真に堅固で安全な透かしの実現が得られるのである。

【００２８】

対照的に、本発明は、様々な暗号化プロトコルを用いて実現し、基礎となるシステムにおける信頼性及びセキュリティの両方を強化することができる。所定の鍵は、マスクの組として説明される。これらのマスクには、基本、畳み込み及びメッセージ・デリミタが含まれるが、メッセージのデジタル署名などの追加的な領域にも拡張することができる。これまでに開示されている技術では、これらのマスクの機能は，写像に対してだけ定義されていた。公開及び秘密鍵を鍵の対として用いて，鍵が危険にさらされることがない可能性を増加させることができる。符号化の前に、上述のマスクは、暗号的な見地から安全なランダム発生プロセスによって発生される。ＤＥＳなどのブロック暗号は、十分にランダムなシード値（seed value）と組み合わされて、暗号的に安全なランダム・ビット発生器をエミュレートする。これらの鍵は、考察しているサンプル・ストリームにそれら

を一致させる情報と共にデータベースにセーブされ、デスクランブリング（スクランブル解除）や後の検出又は復号化動作に用いられる。

【００２９】

これらの同じ暗号化プロトコルを、スクランブルされていない状態でストリームされたコンテンツを正しく表示又は再生するために認証された鍵を要求するストリームされたコンテンツを管理する際に、本発明の実施例と組み合わせることができる。デジタル透かしの場合と同様に、対称的又は非対称的な公開鍵の対が、様々な実現例において用いられる。更に、真正の鍵の対を維持する認証機関に対する必要性も、対称的な鍵の実現例以上のセキュリティを得るためには、伝送の際のセキュリティを考える際には考慮すべき問題となる。

【００３０】

次に、本発明によるデジタル情報保護システムの或る実施例を説明する。ここで添付の図面を参照するが、同じ要素については、複数の図面にわたって同じ参照番号が付されている。図１には、本発明の実施例によるデジタル情報符号化方法のブロック流れ図が図解されている。１つの画像が「ブロック」ごとに処理されるのであるが、ここで、各ブロックは、例えば、単色チャネルにおける３２Ｘ３２のピクセル領域である。ステップ１１０では、各ブロックが、スペクトル変換又は高速フーリエ変換（ＦＦＴ）を用いて、周波数領域に変換される。ステップ１２０及び１３０において、最大の３２の振幅が識別され、これら３２の中の部分集合が、鍵からの基本マスクを用いて選択される。次に、１メッセージ・ビットが、ステップ１４０及び１５０において各ブロックの中に符号化される。このビットは、畳み込みマスクを用いて発生された変換テーブルを用いてメッセージから選ばれる。このビットが真である場合には、選択された振幅は、ユーザによって定義された強度率（strength fraction）だけ減少される。ビットが偽である場合には、振幅は不変である。

【００３１】

選択された振幅と周波数とは、それぞれが、鍵の中に記憶される。すべての画像が処理された後で、ピクセルの対角線方向のストライプが鍵にセーブされる。このストライプは、例えば、左上の角で開始して、画像を通って４５度の角度で

進むことができる。画像の元の寸法も、鍵に記憶される。

【００３２】

図２は、本発明の実施例によるデジタル情報デスケーリング方法のブロック流れ図である。画像が復号化のために選ばれると、最初に、クロッピング及び／又はスケーリングがなされているかどうかがチェックされる。されている場合には、画像は、ステップ２１０において、元の寸法にスケーリングされる。結果的に得られる「ストライプ」すなわちピクセルの対角線は、ステップ２２０において、鍵に記憶されているストライプとの適合が調べられる。適合がそれ以前の最良の適合よりも優れている場合には、スケールがステップ２３０及び２４０においてセーブされる。望むのであれば、例えば、ステップ２６０において、ゼロ・ピクセルの単一のロー又はコラムを用いて、画像をパディングすることができる。そして、このプロセスを反復して、適合が改善するかどうかを見ることができる。

【００３３】

ステップ２５０において完全な適合が見出される場合には、プロセスは終了する。完全な適合が得られない場合には、ユーザによって設定されるクロップ「半径」まで、プロセスが継続される。例えば、クロップ半径が４である場合には，画像を、４つのロー及び／又は４つのコラムまでパディングすることができる。ゼロによって置き換えられた任意のクロッピングされた領域を用いて、最良の適合が選ばれ、画像は、園もとの寸法まで回復される。

【００３４】

情報は、いったんデスケーリングされると、図３に示されている本発明の実施例に従って復号化される。復号化は、符号化の逆プロセスである。復号化された振幅は、鍵に記憶されたものと比較され、ステップ３１０及び３２０において、符号化されたビットの位置が決定される。メッセージは、ステップ３３０において、逆変換テーブルを用いてアセンブルされる。次に、ステップ３４０では、メッセージはハッシュ化され、このハッシュが元のメッセージのハッシュと比較される。元のハッシュは、符号化の間に鍵に記憶される。ハッシュが一致する場合には、メッセージは有効であると宣言され、ステップ３５０においてユーザに与

えられる。

【００３５】

この出願においては様々な実施例が特に図解され説明されているが、本発明の修正及び変形は、以上の説明によってカバーされ、本発明の精神と意図された範囲とから逸脱することなく、冒頭の特許請求の範囲に含まれる。更に、オーディオ及びビデオ・コンテンツに対して、時間ベースの信号操作や振幅及びピッチ動作のために、同様の動作が適用された。透かしの入れられていないオリジナルを用いることなくデスケーリング又はそれ以外の態様で迅速に差異を判断できる能力が、安全なデジタル透かしにとっては、固有の重要性を有している。デジタル化されたコンテンツはネットワークを介して交換されるので、拒絶されないことと第三者による認証とを保証することも重要である。

【図面の簡単な説明】

【図１】

本発明の或る実施例によるデジタル情報の符号化方法のブロック流れ図である。

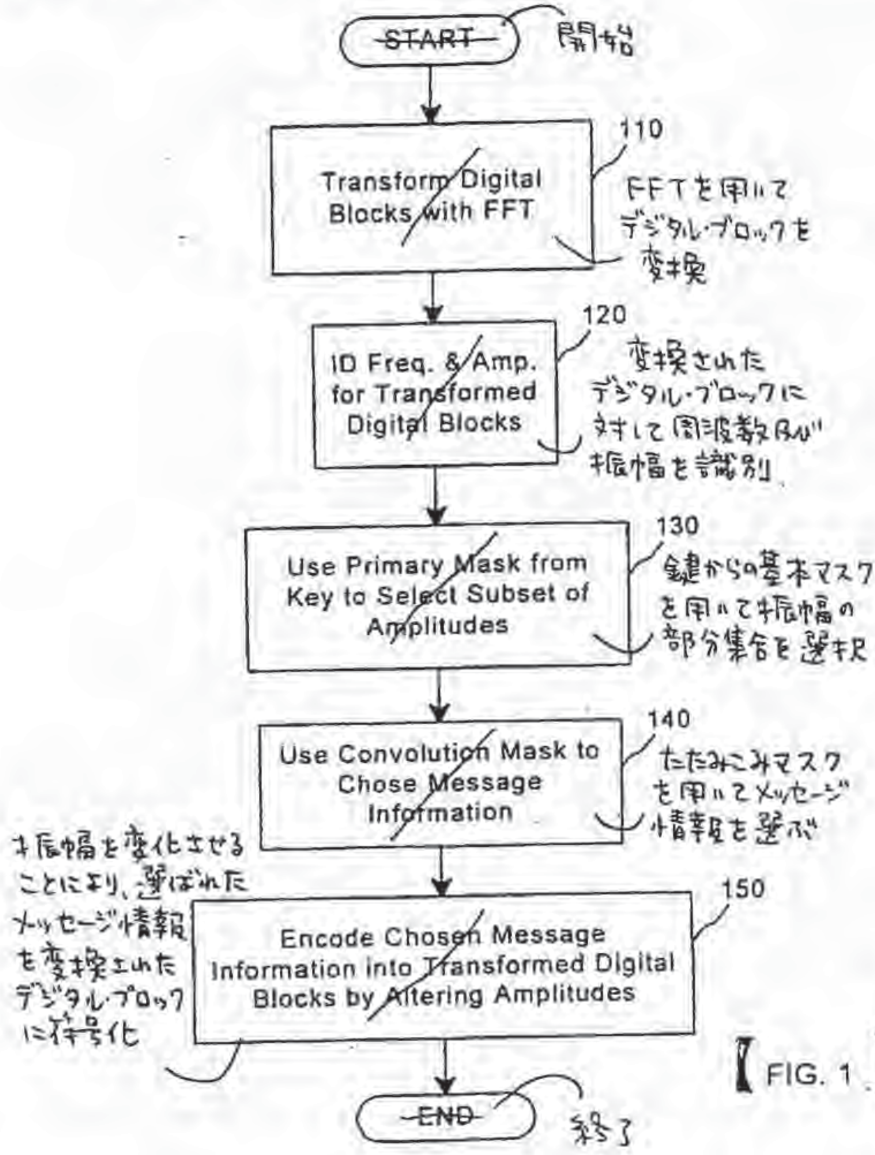
【図２】

本発明の或る実施例によるデジタル情報のデスケーリング方法のブロック流れ図である。

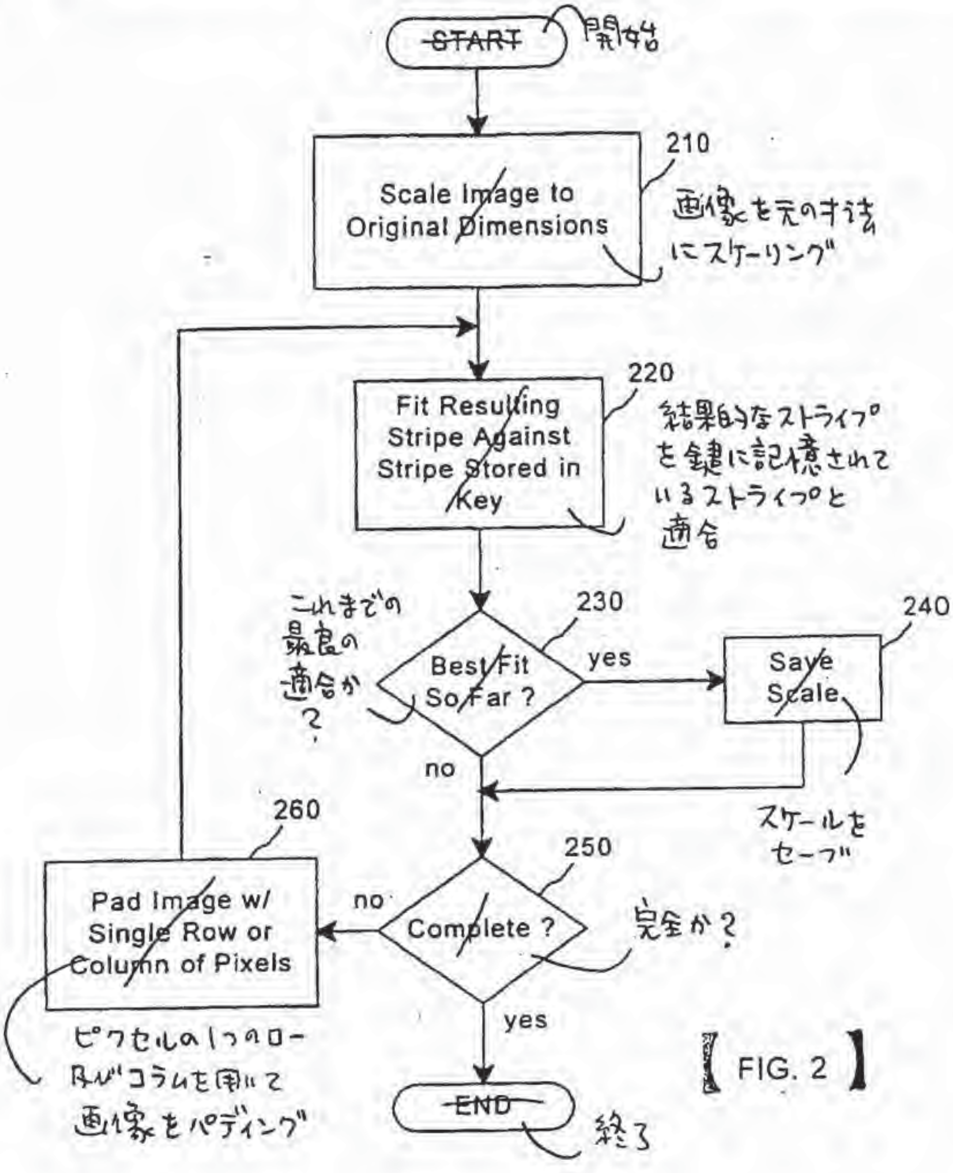
【図３】

本発明の或る実施例によるデジタル情報の復号化方法のブロック流れ図である。

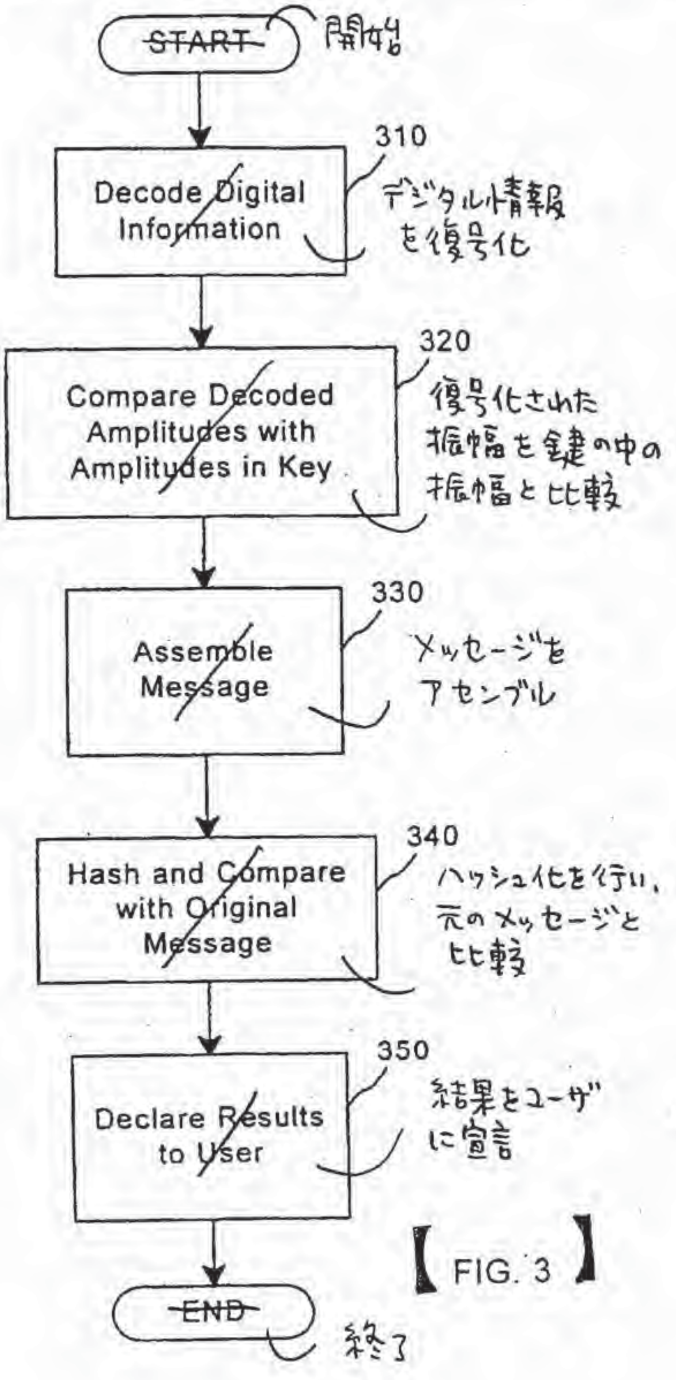
【書類名】　図面


~~START~~
開始
110
Transform Digital Blocks with FFT
FFTを用いてデジタル・ブロックを変換
120
ID Freq. & Amp. for Transformed Digital Blocks
変換されたデジタル・ブロックに対して周波数及び振幅を識別
130
Use Primary Mask from Key to Select Subset of Amplitudes
鍵からの基本マスクを用いて振幅の部分集合を選択
140
Use Convolution Mask to Chose Message Information
たたみこみマスクを用いてメッセージ情報を選ぶ
150
Encode Chosen Message Information into Transformed Digital Blocks by Altering Amplitudes
振幅を変化させることにより、選ばれたメッセージ情報を変換されたデジタル・ブロックに符号化
~~END~~
終了

【FIG. 1】


~~START~~
開始
210
Scale Image to Original Dimensions
画像を元の寸法にスケーリング
220
Fit Resulting Stripe Against Stripe Stored in Key
結果的なストライプを鍵に記憶されているストライプと適合
230
Best Fit So Far ?
これまでの最良の適合か？
yes
no
240
Save Scale
スケールをセーブ
250
Complete ?
完全か？
no
yes
260
Pad Image w/ Single Row or Column of Pixels
ピクセルの1つのロー及びコラムを用いて画像をパディング
~~END~~
終了

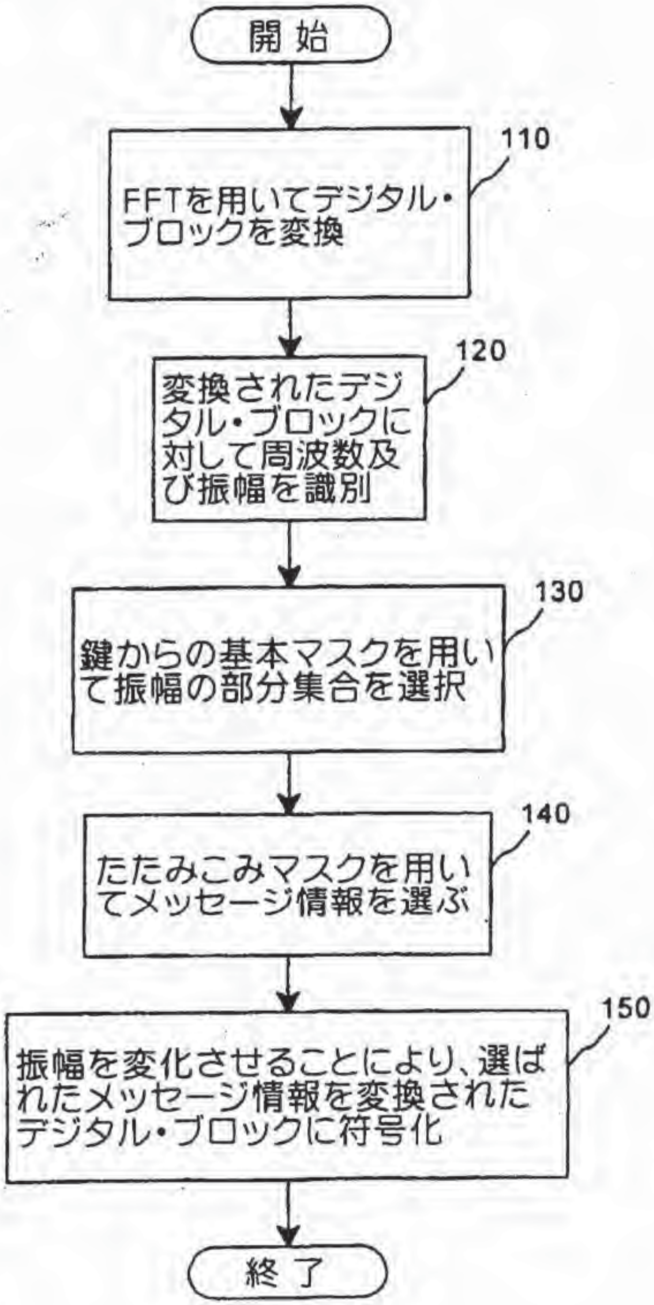
【 FIG. 2 】


~~START~~
開始
310
~~Decode Digital Information~~
デジタル情報を復号化
320
~~Compare Decoded Amplitudes with Amplitudes in Key~~
復号化された振幅を鍵の中の振幅と比較
330
~~Assemble Message~~
メッセージをアセンブル
340
~~Hash and Compare with Original Message~~
ハッシュ化を行い、元のメッセージと比較
350
~~Declare Results to User~~
結果をユーザに宣言
~~END~~
終了

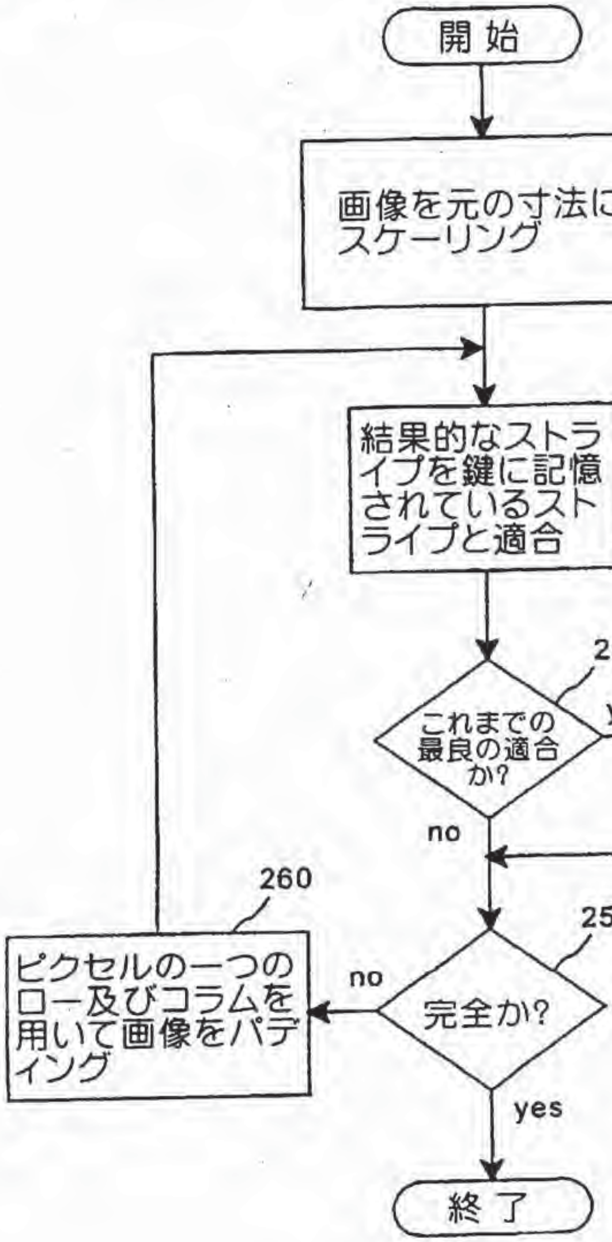
【FIG. 3】

【書類名】　図面

【図１】


開始
110
FFTを用いてデジタル・ブロックを変換
120
変換されたデジタル・ブロックに対して周波数及び振幅を識別
130
鍵からの基本マスクを用いて振幅の部分集合を選択
140
たたみこみマスクを用いてメッセージ情報を選ぶ
150
振幅を変化させることにより、選ばれたメッセージ情報を変換されたデジタル・ブロックに符号化
終了

【図２】


開 始
210
画像を元の寸法にスケーリング
220
結果的なストライプを鍵に記憶されているストライプと適合
230
これまでの最良の適合か?
yes
240
スケールをセーブ
no
250
完全か?
no
260
ピクセルの一つのロー及びコラムを用いて画像をパディング
yes
終 了

【図３】


開始
310
デジタル情報を復号化
320
復号化された振幅を鍵の中の振幅と比較
330
メッセージをアセンブル
340
ハッシュ化を行い、元のメッセージと比較
350
結果をユーザに宣言
終了

【書類名】　要約書

【要約】　安全なデジタル透かしのための複数の変換の利用及び適用である。本発明の或る実施例では、保護されるべきデジタル情報におけるデジタル・ブロックは、高速フーリエ変換を用いて周波数領域に変換される。複数の周波数及び関連する振幅が、変換されたデジタル・ブロックのそれぞれに対して識別され、識別された振幅の部分集合が、鍵からの基本マスクを用いてデジタル・ブロックのそれぞれに対して選択される。メッセージ情報が、畳み込みマスクを用いて発生された変換テーブルを用いて、メッセージから選択される。選ばれたメッセージ情報は、選択されたメッセージ情報に基づいて選択される振幅を変化させることによって、変換されたデジタル・ブロックのそれぞれに符号化される。

Your Ref.: 066358.0102JP
(Our Ref.: EL:TOT:MOH, S-1181-1/002365)

**Amendment**

提出日　平成１２年１０月１３日
整理番号＝００２３６５１　PCT/US99/07262　頁：　1/　1

Filed: October 13, 2000

【書類名】　手続補正書
【整理番号】　0023651
【提出日】　平成12年10月13日
【あて先】　特許庁長官　殿
【事件の表示】
　　【国際出願番号】　PCT/US99/07262
　　【出願の区分】　特許
【補正をする者】
　　【住所又は居所】　アメリカ合衆国フロリダ州３３１６０，マイアミ，コリンズ・アベニュー　１６７１１，ナンバー　２５０５
　　【氏名又は名称】　スコット・エイ・モスコウィッツ
【代理人】
　　【識別番号】　100089705
　　【住所又は居所】　東京都千代田区大手町二丁目２番１号　新大手町ビル２０６区　ユアサハラ法律特許事務所
　　【弁理士】
　　【氏名又は名称】　社本　一夫
【手続補正　1】
　【補正対象書類名】　図面
　【補正対象項目名】　全図
　【補正方法】　変更
　【補正の内容】　1
【その他】　浄書につき、図面の実体的内容には変更なし。
【プルーフの要否】　要

# VOLLMACHT[1] / AUTHORISATION[1] / POUVOIR[1]

Bitte vor dem Ausfüllen des Formblatts Rückseite beachten
Please read the notes overleaf before completing the form
Veuillez lire les remarques au verso avant de remplir le formulaire

Zeichen des Vertreters (der Vertreter) (max. 15 Positionen) / Representative's Reference (max. 15 spaces) / Référence du (des) mandataire(s) (15 caractères ou espaces au maximum)

Nr. der Anmeldung (des Patents) / Application/Patent No. / N° de la demande (du brevet)

Ich (Wir) / I (We) / Je (Nous)[2]

THE DICE COMPANY
16711 Collins Avenue #2505
Miami, Florida 33160
USA

bevollmächtige(n) hiermit / do hereby authorise / autorise (autorisons) par la présente[3]

☐ (Weitere Vertreter sind auf einem gesonderten Blatt angegeben. / Additional representatives indicated on supplementary sheet. / Les autres mandataires sont mentionnés sur une feuille supplémentaire.)

mich (uns) zu vertreten als / to represent me (us) as / à me (nous) représenter en tant que

☒ Anmelder oder Patentinhaber / applicant(s) or patent proprietor(s) / demandeur(s) ou titulaire(s) du brevet.

☐ Einsprechenden (Einsprechende) / opponent(s) / opposant(s).

für mich (uns) zu handeln in den durch das Europäische Patentübereinkommen geschaffenen Verfahren in der (den) folgenden europäischen Patentanmeldung(en) oder dem (den) folgenden europäischen Patent(en)[4] und Zahlungen für mich (uns) in Empfang zu nehmen:
to act for me (us) in all proceedings established by the European Patent Convention concerning the following European patent application(s) or patent(s)[4] and to receive payments on my (our) behalf:
à agir en mon (notre) nom dans toute procédure instituée par la Convention sur le brevet européen et concernant la (les) demande(s) de brevet ou le (les) brevet(s) européen(s)[4] suivant(s) et à recevoir des paiements en mon (notre) nom:

Regional Phase of PCT/US96/10257 for STEGANOGRAPHIC METHOD AND DEVICE

☒ Weitere Anmeldungen oder Patente sind auf einem gesonderten Blatt angegeben. / Additional applications or patents indicated on supplementary sheet. / Les autres demandes ou brevets sont mentionnés sur une feuille supplémentaire.

☒ Die Vollmacht gilt auch für Verfahren nach dem Vertrag über die internationale Zusammenarbeit auf dem Gebiet des Patentwesens. This authorisation shall also apply to the same extent to any proceedings established by the Patent Cooperation Treaty. Ce pouvoir s'applique également à toute procédure instituée par le Traité de coopération en matière de brevets.

☒ Diese Vollmacht gilt auch für eventuelle europäische Teilanmeldungen. / This authorisation also covers any European divisional applications. / Le présent pouvoir vaut également pour les demandes divisionnaires européennes qui pourraient être déposées.

☒ Untervollmacht kann erteilt werden. / Sub-authorisation may be given. / Le pouvoir pourra être délégué.

☒ Ich (Wir) widerrufe(n) hiermit frühere Vollmachten in Sachen der obenbezeichneten Anmeldung(en) oder des obenbezeichneten Patents (der obenbezeichneten Patente)[5]. / I (We) hereby revoke all previous authorisations in respect of the above application(s) or patent(s)[5]. / Je révoque (Nous révoquons) par la présente tout pouvoir antérieur, donné pour la (les) demande(s) ou le (les) brevet(s) mentionné(s) ci-dessus[5].

Ort / Place / Lieu: Miami, Florida

Datum / Date: 12-8-97

Unterschrift(en) / Signature(s)[6]

Scott Moskowitz

Scott A. Moskowitz, President

Das Formblatt muß vom (von den) Vollmachtgeber(n) (bei juristischen Personen vom Unterschriftsberechtigten) eigenhändig unterzeichnet sein. Nach der Unterschrift bitte den (die) Namen des (der) Unterzeichneten mit Schreibmaschine wiederholen (bei juristischen Personen die Stellung des Unterschriftsberechtigten innerhalb der Gesellschaft angeben).

The form must bear the personal signature(s) of the authorisor(s) (in the case of legal persons, that of the officer empowered to sign). After the signature, please type the name(s) of the signatory(ies) adding, in the case of legal persons, his (their) position within the company.

Le formulaire doit être signé de la propre main du (des) mandant(s) (dans le cas de personnes morales, de la personne ayant qualité pour signer). Veuillez ajouter à la machine, après la signature, le (les) nom(s) du (des) signataire(s) en mentionnant, dans le cas de personnes morales, ses (leurs) fonctions au sein de la société.

EPA/EPO/OEB Form 1003 10.88

bitte wenden / P.T.O. / T.S.V.P.

**PCT**

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

| | | |
|---|---|---|
| **(51) International Patent Classification** $^6$ :<br>**H04L** | **A2** | **(11) International Publication Number:** **WO 96/42151**<br>**(43) International Publication Date:** 27 December 1996 (27.12.96) |

**(21) International Application Number:** PCT/US96/10257

**(22) International Filing Date:** 7 June 1996 (07.06.96)

**(30) Priority Data:**
08/489.172 9 June 1995 (09.06.95) US

**(71) Applicant:** THE DICE COMPANY [US/US]; P.O. Box 60471, Palo Alto, CA 94306-0471 (US).

**(72) Inventors:** COOPERMAN, Marc, S.; 2929 Ramona, Palo Alto, CA 94306 (US). MOSKOWITZ, Scott, A.; Townhouse 4, 20191 East Country Club Drive, North Miami Beach, FL 33180 (US).

**(74) Agents:** ALTMILLER, John, C. et al.; Kenyon & Kenyon, 1025 Connecticut Avenue, N.W., Washington, DC 20036 (US).

**(81) Designated States:** CA, CN, FI, JP, KR, SG, European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

**Published**
*Without international search report and to be republished upon receipt of that report.*

**(54) Title:** STEGANOGRAPHIC METHOD AND DEVICE

**(57) Abstract**

An apparatus and method for encoding and decoding additional information into a stream of digitized samples in an integral manner. The information is encoded using special keys. The information is contained in the samples, not prepended or appended to the sample stream. The method makes it extremely difficult to find the information in the samples if the proper keys are not possessed by the decoder. The method does not cause a significant degradation to the sample stream. The method is used to establish ownership of copyrighted digital multimedia content and provide a disincentive to piracy of such material.

## *FOR THE PURPOSES OF INFORMATION ONLY*

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

| | |
|---|---|
| AM | Armenia |
| AT | Austria |
| AU | Australia |
| BB | Barbados |
| BE | Belgium |
| BF | Burkina Faso |
| BG | Bulgaria |
| BJ | Benin |
| BR | Brazil |
| BY | Belarus |
| CA | Canada |
| CF | Central African Republic |
| CG | Congo |
| CH | Switzerland |
| CI | Côte d'Ivoire |
| CM | Cameroon |
| CN | China |
| CS | Czechoslovakia |
| CZ | Czech Republic |
| DE | Germany |
| DK | Denmark |
| EE | Estonia |
| ES | Spain |
| FI | Finland |
| FR | France |
| GA | Gabon |
| GB | United Kingdom |
| GE | Georgia |
| GN | Guinea |
| GR | Greece |
| HU | Hungary |
| IE | Ireland |
| IT | Italy |
| JP | Japan |
| KE | Kenya |
| KG | Kyrgystan |
| KP | Democratic People's Republic of Korea |
| KR | Republic of Korea |
| KZ | Kazakhstan |
| LI | Liechtenstein |
| LK | Sri Lanka |
| LR | Liberia |
| LT | Lithuania |
| LU | Luxembourg |
| LV | Latvia |
| MC | Monaco |
| MD | Republic of Moldova |
| MG | Madagascar |
| ML | Mali |
| MN | Mongolia |
| MR | Mauritania |
| MW | Malawi |
| MX | Mexico |
| NE | Niger |
| NL | Netherlands |
| NO | Norway |
| NZ | New Zealand |
| PL | Poland |
| PT | Portugal |
| RO | Romania |
| RU | Russian Federation |
| SD | Sudan |
| SE | Sweden |
| SG | Singapore |
| SI | Slovenia |
| SK | Slovakia |
| SN | Senegal |
| SZ | Swaziland |
| TD | Chad |
| TG | Togo |
| TJ | Tajikistan |
| TT | Trinidad and Tobago |
| UA | Ukraine |
| UG | Uganda |
| US | United States of America |
| UZ | Uzbekistan |
| VN | Viet Nam |

**PCT**

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

# INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

| | | |
|---|---|---|
| **(51) International Patent Classification** $^6$ :<br>**H04L** | **A2** | **(11) International Publication Number:** **WO 96/42151**<br>**(43) International Publication Date:** 27 December 1996 (27.12.96) |

**(21) International Application Number:** PCT/US96/10257

**(22) International Filing Date:** 7 June 1996 (07.06.96)

**(30) Priority Data:**
08/489.172 9 June 1995 (09.06.95) US

**(71) Applicant:** THE DICE COMPANY [US/US]; P.O. Box 60471, Palo Alto, CA 94306-0471 (US).

**(72) Inventors:** COOPERMAN, Marc, S.; 2929 Ramona, Palo Alto, CA 94306 (US). MOSKOWITZ, Scott, A.; Townhouse 4, 20191 East Country Club Drive, North Miami Beach, FL 33180 (US).

**(74) Agents:** ALTMILLER, John, C. et al.; Kenyon & Kenyon, 1025 Connecticut Avenue, N.W., Washington, DC 20036 (US).

**(81) Designated States:** CA, CN, FI, JP, KR, SG, European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

**Published**
*Without international search report and to be republished upon receipt of that report.*

**(54) Title:** STEGANOGRAPHIC METHOD AND DEVICE

**(57) Abstract**

An apparatus and method for encoding and decoding additional information into a stream of digitized samples in an integral manner. The information is encoded using special keys. The information is contained in the samples, not prepended or appended to the sample stream. The method makes it extremely difficult to find the information in the samples if the proper keys are not possessed by the decoder. The method does not cause a significant degradation to the sample stream. The method is used to establish ownership of copyrighted digital multimedia content and provide a disincentive to piracy of such material.

# STEGANOGRAPHIC METHOD AND DEVICE

## Definitions

Several terms of art appear frequently in the following. For ease of reference they are defined here as follows:

"Content" refers to multimedia content. This term encompasses the various types of information to be processed in a multimedia entertainment system. Content specifically refers to digitzed audio, video or still images in the context of this discussion. This information may be contained within files on a multimedia computer system, the files having a particular format specific to the modality of the content (sound, images, moving pictures) or the type of systems, computer or otherwise, used to process the content.

"Digitized" refers to content composed of discrete digital samples of an otherwise analog media, which approximate that media inside a computer or other digital device. For instance, the sound of music occurs naturally, and is experienced by humans as an analog (continuous) sound wave. The sound can be digitized into a stream of discrete samples, or numbers, each of which represents an approximate value of the amplitude of the real analog wave at a particular instant in time. These samples can be stored in files in a computer and then used to recreate the original sound wave to a high degree of accuracy.

In general, content entering a digital system is digitized by Analog to Digital converters (A/D) and analog media are recreated by the digital system using a Digital to Analog (D/A) converter. In the context of this discussion content is always digitized content.

"Cryptography" is a field covering numerous techniques for scrambling information conveying messages so that when the message is conveyed between the sender and receiver an unintended party who intercepts this message cannot read it, or extract useful information from it.

A "Public Key Cryptosystem" is a particular cryptographic system where all parties possess pairs of keys for encryption and decryption. Parties to this type of system freely distribute their public keys, which other may use to encrypt messages to the owner of the public key. Such messages are decrypted by the receiver with the private key. Private keys are never distributed. A message encrypted with a public key can only be decrypted with the corresponding private key, and vice versa. A message encrypted with a private key is said to have been signed by the owner of that key. Anyone in possession of the public key may decrypt the message and know that it was encrypted, and thus signed, by the owner of the public key, since only they possess the corresponding private key.

"Steganography" is a field distinguished from cryptography, but associated with it, that covers numerous methods for hiding an informational message within some other medium, perhaps another unrelated message, in such a manner that an unintended party who intercepts the medium carrying the hidden message does not know it contains this hidden message and therefore does not obtain the information in the hidden message. In other words, steganography seeks to hide messages in plain view.

## Background of the Invention

In the current environment of computer networks and the proliferation of digital or digitized multimedia content which may be distributed over such networks, a key issue is copyright protection. Copyright protection is the ability to prevent or deter the proliferation of unauthorized copies of copyrighted works. It provides a reasonable guarantee that the author of a copyrighted work will be paid for each copy of that work.

A fundamental problem in the digital world, as opposed to the world of physical media, is that a unlimited number of perfect copies may be made from any piece of digital or digitized content. A perfect copy means that if the original is comprised of a given stream of numbers, then the copy matches the original, exactly, for each number in the stream. Thus, there is no degradation of the original signal during the copy operation. In an analog copy, random noise is always introduced, degrading the copied signal.

The act of making unlicensed copies of some content, digital or analog, whether audio, video, software or other, is generally known as *piracy*. Piracy has been committed for the purpose of either profit from the sale of such unlicensed copies, or to procure for the "pirate" a copy of the content for personal use without having paid for it.

The problem of piracy has been made much worse for any type of content by the digitization of content. Once content enters the digital domain, an unlimited number of copies may be made without any degradation, if a pirate finds a way to break whatever protection scheme was established to guard against such abuses, if any. In the analog world, there is generally a degradation in the content (signal) with each successive copy, imposing a sort of natural limit on volume of piracy.

To date, three general types of schemes have been implemented in an attempt to protect copyrights.

1) Encryption
2) Copy Protection
3) Content Extensions

Copy Protection and Content Extensions generally apply in the digital world only, while a scheme related to Encryption, commonly known as scrambling, my be applied to an analog signal. This is typical in analog cable systems.

**Encryption** scrambles the content. Before the content is made ready for delivery, whether on floppy disk, or over a network, it must be encrypted, or scrambled. Once the content has been encrypted, it cannot be used until it is decrypted, or unscrambled. Encrypted audio data might sound like incomprehensible screeching, while an encrypted picture or video might appear as random patterns on a screen. The principle of encryption is that you are free to make as many copies as you want, but you can't read anything that makes sense until you use a special key to decrypt, and you can only obtain the key by paying for the content.

Encryption has two problems, however. 1) Pirates have historically found ways to crack encryption, in effect, obtaining the key without having paid for it; and 2) Once a single legitimate copy of some content has been decrypted, a pirate is now free to make unlimited copies of the decrypted copy. In effect, in order to sell an unlimited quantity of an encrypted piece of software, the pirate could simply buy one copy, which they are entitled to decrypt.

**Copy Protection** includes various methods by which a software engineer can write the software in a clever manner to determine if it has been copied, and if so to deactivate itself. Also included are undocumented changes to the storage format of the content. Copy protection was generally abandoned by the software industry, since pirates were generally just as clever as the software engineers and figured out ways to modify their software and deactivate the protection. The cost of developing such protection was not justified considering the level of piracy which occurred despite the copy protection.

**Content Extension** refers to any system which attaches some extra information to the original content which indicates whether or not a copy may be made. A software or hardware system must be specifically built around this scheme to recognize the additional information and interpret it in an appropriate manner. An example of such a system is the Serial Copyright Management System embedded in Digital Audio Tape (DAT) hardware. Under this system, additional information is stored on the disc immediately preceding each track of audio content which indicates whether or not it can be copied. The hardware reads this information and uses it accordingly.

A fundamental problem with Encryption and Content Extension is the "rogue engineer". An employee who helped design such a system or an individual with the knowledge and means to analyze such a system can modify it to ignore the copyright information altogether, and make unlicensed copies of the content. Cable piracy is quite common, aided by illicit decoder devices built by those who understand the technical details of the cable encryption system. Although the cable systems in question were actually based on analog RF signals, the same principle applies to digital systems.

The practical considerations of weak encryption schemes and rogue engineers have served to limit the faith which may be put in such copyright protection schemes. The invention disclosed herein serves to address these problems with conventional systems for digital distribution. It provides a way to enforce copyright online. The invention draws on techniques from two fields, cryptography, the art of scrambling messages so that only the intended recipient may read them, and steganography, a term applied to various techniques for obscuring messages so that only the intended parties to a message even know that a message has been sent, thus it is termed herein as a stega-cipher. The stega-cipher is so named because it uses the steganographic technique of hiding a message in multimedia content, in combination with multiple keys, a concept originating in cryptography. However, instead of using the keys to encrypt the content, the stega-cipher uses these kevs to locate the hidden message within the content. The message itself is encrypted which serves to further protect the message, verify the validity of the message, and redistribute the information in a random manner so that anyone attempting to locate the message without the keys cannot rely on pre-supposed knowledge of the message contents as a help in locating it.

## Summary of the Invention

The invention disclosed herein combines two techniques, steganography - obscuring information that is otherwise in plain sight, and cryptography - scrambling information that must be sent over unsecured means, in a manner such that only the intended recipient may successfully unscramble it. The net effect of this system is to specifically watermark a piece of content so that if it is copied, it is possible to determine who owned the original from which the copies were made, and hence determine responsibility for the copies. It is also a feature of the system to uniquely identify the content to which it is applied.

For a comprehensive discussion of cryptography, its theory, applications and specific algorithms, see APPLIED CRYPTOGRAPHY, by Bruce Schneier, which is herein incorporated by reference at pages 66-68, 387-392.

Steganography is discussed briefly in THE CODE BREAKERS by David Kahn, which is herein incorporated by reference at pages xiii, 81-83, 522-526, and 873. An example application, Stego by Romana Machado, is also available for the Apple Macintosh. Stego can be found at the internet uniform resource locator "ftp://sumex-aim.stanford.edu/info-mac/cmp/stego10a2.hqx". This application demonstrates a simple steganographic technique to encode a text message into a graphical image without significantly distorting the image.

The invention improves upon the prior art by providing a manner for protecting copyright in the digital domain, which neither steganography or cryptography does. It improves specifically on steganography by making use of special keys which dictate exactly where within a larger chunk of content a message is to be hidden, and makes the task of extracting such a message without the proper key the equivalent of looking for a needle in a haystack.

The information encoded by the Stega-Cipher process serves as a watermark which identifies individual copies of content legally licensed to specific parties. It is integral with the content. It cannot be removed by omission in a transmission. It does not add any overhead to signal transmission or storage. It does allow the content to be stored to and used with traditional offline analog and digital media, without modification or significant signal degradation. These aspects of the stega-cipher all represent improvements to the art. That is, its forces would - be pirates to damage the content in order to guarantee the disabling of the watermark.

The invention described herein is used for protecting and enforcing copyrights in the digital or on-line domain, where there are no physical limitations on copying copyrighted content.

The invention uniquely identifies every copy of multimedia content made using the invention, composed of digitized samples whether compressed or uncompressed, including but not limited to still digital images, digital audio, and digital video.

The invention is for use in meterware or pay-by-use systems where an online user incurs a charge each time they access a particular piece of content, or uses a software title.

The invention is for use as a general improvement to cryptographic techniques to increase the complexity of cryptanalysis on a given cipher.

It is considered that the method and steps of the present invention will be modified to account for the effects of loss compression schemes on the samples and particularly includes modification to handle MPEG compressed audio and video.

It is considered that statistical data spreading and recovery techniques, error coding or spread spectrum processing techniques might be applied in the invention to handle the effects of loss compression, or counter the effects of a randomization attack.

It is considered that the apparatus described might be further specialized and optimized in hardware by replacing general purpose data buses and CPU or DSP driven operations with hardwired circuitry, incorporated in one or more special purpose ICs.

It is considered that the apparatus will be modeled and implemented in software on general purpose computer platforms.

It is considered that stega-cipher hardware could be embedded in a consumer electronics device and used to not only identify content and copyright, but to enable use of that content.

## Detailed Description

### I. Digital Copyright Stega-Cipher Protocol and the Decode/Encode Program

The purpose of the program described here is to watermark digital multimedia content for distribution to consumers through online services in such a way as to meet the following criteria

Given a unique piece of multimedia content, composed of digitized samples, it is desirable to:

1) Uniquely identify this particular piece of content from others in a manner which is secure and undeniable (e.g. to know whether a digital audio recording is "My Way" by Frank Sinatra, or "Stairway to Heaven", by Led Zeppelin), and in a manner such that this identification can be performed automatically by an electronic device or mechanism.

2) Uniquely identify the copyright owner of the content, and the terms under which it may be distributed in general, in a manner which is secure and undeniable.

3) At such time as is necessary, additionally, uniquely identify in a secure and undeniable manner the licensed publisher who received a particular copy of the content, and the terms under which they may redistribute or resell it.

4) At such time as is necessary, additionally, uniquely identify in a secure and undeniable manner, the licensed subscriber who received a particular copy of the content from the publisher described in item 3.

The program described in more detail below combines the techniques of cryptography and steganography to hide a securely encrypted digital copyright certificate which contains information satisfying the criteria listed above, in such a manner as to be integral with the content, like a watermark on paper, so that possession of the content dictates possession of the watermark information. In addition, the watermark cannot be "found" or successfully decoded, without possession of the correct "masks" or keys, available only to those legitimately authorized, namely, those parties to a commercial transaction involving the sale of a copy of the content. Finally, the ability to distribute such watermarked content in a system which implements the watermark scheme is denied without a successfully decoded watermark. Because well known and tested cryptographic techniques are used to protect the certificate itself, these certificates are virtually impossible to forge. Finally, the watermark cannot be erased without significantly damaging the content.

The basic program represents a key part of the invention itself. This program is then used as the method by which copyright information is to be associated in an integral manner with the content. This is a concept absent from copy protection, encryption and content extension schemes. The copyright information itself can be made undeniable and unforgeable using cryptographic techniques, so that through it an audit trail of ownership my be established for each copy of a given piece of content, thus customizing each copy to a particular owner, in a way that can be used to identify the owner.

The value of the stega-cipher is that it provides a way to watermark the content in a way that changes it slightly, but does not impact human perception significantly. And, furthermore, that it is made difficult to defeat since one must know exactly where the information resides to extract it for analysis and use in forgery attempts, or to remove it without overly degrading the signal. And, to try to forge copyright information one must first be able to analyze the encrypted copyright information, and in order to do that, one must be able to find it, which requires masks.

## II. Example Embodiment of General Processing

Digital audio data is represented by a series of samples in 1 dimension,

$$\{S_1, S_2, S_3 \ldots S_n\}$$

This series is also referred to as a sample stream. The sample stream approximates an analog waveform of sound amplitude over time. Each sample represents an estimate of the wave amplitude at the instant of time the sample is recorded. For monaural audio, there is one such sample stream. Stereo audio is comprised of two sample streams, one representing the right channel, and the other representing the left. Each stream is used to drive a corresponding speaker to reproduce the stereo sound.

What is referred to as CD quality audio is characterized by 16 bit (2 byte) stereo samples, recorded at 44.1 Khz, or 44,100 samples per second in each channel. The dynamic range of sound reproduction is directly proportional to the number of bits per sample. Some lower quality recordings are done at 8 bits. A CD audio recording can be stored using any scheme for containing the 2 sample streams in their entirety. When these streams are played back at the same frequency they were recorded at, the sound recorded is reproduced to a high degree of accuracy.

The sample stream is processed in order from first sample to last. For the purpose of the invention disclosed, the stream is separated into sample windows, each of which has a fixed number of consecutive samples from the stream, and where windows do not overlap in the sample stream. Windows may be contiguous in the sample stream. In this discussion assume each window contains 128 samples, and that windows are contiguous. So, the windows within the stream look like

$$\{[S_1, S_2, S_3 \ldots S_{128}], [S_{129}, S_{130}, S_{131} \ldots S_{256}], \ldots [S_{n-128} \ldots S_n] \}$$

where [...] denotes each window and any odd samples at the end of the stream which do not completely fill a window can be ignored, and simply passed through the system unmodified.

These windows will be used as input for the discrete Fast Fourier Transform (and its inverse) operation.

Briefly, Fourier Transform methods are based on the principle that a complex waveform, expressed as amplitude over time and represented by a sample stream, is really the sum of a number of simple waveforms, each of which oscillate at different frequencies.

By complex, it is meant that the value of the next sample is not easily predicted from the values of the last N samples or the time of the sample. By simple it is meant that the value of the sample is easily predictable from the values of the last N samples and/or the time of the sample.

The sum of multiple simple waves is equivalent to the complex wave. The discrete FFT and its inverse simply translate a limited amount of data from one side of this equivalence to the other, between the complex waveform and the sum of simple waves. The discrete FFT can be used to translate a series of samples representing amplitude over time (the complex wave, representing a digital audio recording) into the same number of samples representing total spectral energy in a given range of frequencies (the simple wave components) at a particular instant of time. This instant is the time in the middle of the original amplitude/time samples. The inverse discrete FFT translates the data in the other direction, producing the complex waveform, from its simpler parts.

Each 128 sample window will be used as an input to the discrete FFT, resulting in 128 bins representing each of 128 frequency bands, ranging from 0Hz to 22Khz (the Nyquist frequency, or ½ the sampling rate).

Information can be encoded into the audio signal in the frequency domain or in the time domain. In the latter case, no FFT or inverse FFT is necessary. However, encoding in the frequency domain is recommended, since its effects are scattered over the resultant time domain samples, and not easily predicted. In addition, frequency domain encoding makes it more likely that randomization will result in noticeable artifacts in the resultant signal, and therefore makes the stega-cipher more defensible against such attacks. It is in the frequency domain that additional information will be encoded into the audio signal for the purpose of this discussion. Each frequency band in a given time slice can potentially be used to store a small portion of some additional information to be added to the signal. Since these are discrete estimates, there is some room for error which will not significantly effect the perceived quality of the signal, reproduced after modification, by the inverse FFT operation. In effect, intentional changes, which cannot be distinguished from random variations are introduced in the frequency domain, for the purpose of storing additional information in the sample stream. These changes are minimized so as not to adversely affect the perceived quality of the reproduced audio signal, after it has been encoded with additional information in the manner described below. In addition, the location of each of these changes is made virtually impossible to predict, an innovation which distinguishes this scheme from simple steganographic techniques.

Note that this process differs from the Nagata, et al. patents, 4,979,210 and 5,073,925, which encode information by modulating an audio signal in amplitude/time domain. It also differs in that the modulations introduced in the Nagata process (which are at very low amplitude and frequency relative to the carrier wave as to remain inaudible) carry only copy/ don't copy information, which is easily found and circumvented by one skilled in the art. Also, there is no limitation in the stega-cipher process as to what type of information can be encoded into the signal, and there is more information storage capacity, since the encoding process is not bound by any particular frequency of modulation but rather by the number of samples available. The granularity of encoding in the stega-cipher is determined by the sample window size, with potentially 1 bit of space per sample or 128 bits per window (a secure implementation will halve this to 64 bits). In Nagata, et al. the granularity of encoding is fixed by the amplitude and frequency modulation limits required to maintain inaudibility. These limits are relatively low, and therefore make it impractical to encode more than simple copy/ don't copy information using the Nagata process.

## III. Example Embodiment of Encoding and Decoding

A modification to standard steganographic technique is applied in the frequency domain described above, in order to encode additional information into the audio signal.

In a scheme adapted from cryptographic techniques, 2 keys are used in the actual encode and decode process. For the purposes of this invention the keys are referred to as masks. One mask, the primary, is applied to the frequency axis of FFT results, the other mask is applied to the time axis (this will be called the convolution mask). The number of bits comprising the primary mask are equal to the sample window size in samples (or the number of frequency bands computed by the FFT process), 128 in this discussion. The number of bits in the convolution mask are entirely arbitrary. This implementation will assume a time mask of 1024 bits. Generally the larger the key, the more difficult it is to guess.

Prior to encoding, the primary and convolution masks described above are generated by a cryptographically secure random generation process. It is possible to use a block cipher like DES in combination with a sufficiently pseudo-random seed value to emulate a cryptographically secure random bit generator. These keys will be saved along with information matching them to the sample stream in question in a database for use in decoding, should that step become necessary.

Prior to encoding, some additional information to be encoded into the signal is prepared and made available to the encoder, in a bit addressable manner (so that it may be read one bit at a time). If the size of the sample stream is known and the efficiency characteristics of the stega-cipher implementation are taken into account, a known limit may be imposed on the amount of this additional information.

The encoder captures one sample window at a time from the sample stream, in sequential, contiguous order. The encoder tracks the sequential number of each window it acquires. The first window is 0. When the number of windows processed reaches the number of bits in the window mask, minus one, the next value of the window counter will be reset to 0.

This counter is the convolution index or phase. In the current implementation it is used as a simple index into the convolution bitmask. In anticipated developments it will be used to perform convolution operations on the convolution mask to determine which bit to use. For instance the mask might by rotated by a number corresponding to the phase, in bits to the left and XORed with the primary mask to produce a new mask, which is then indexed by the phase. There are many possibilities for convolution.

The encoder computes the discrete FFT of the sample window.

Starting with the lowest frequency band, the encoder proceeds through each band to the highest, visiting each of the 128 frequency bands in order. At each band value, the encoder takes the bit of the primary mask corresponding to the frequency band in question, the bit of the convolution mask corresponding to the window in question, and passes these values into a boolean function. This function is designed so that it has a near perfectly random output distribution. It will return true for approximately 50% of its input permutations, and false for the other 50%. The value returned for a given set of inputs is fixed, however, so that it will always return the same value given the same set of inputs.

If the function returns true, the current frequency band in the current window is used in the encoding process, and represents a valid piece of the additional information encoded in the signal. If the function returns false, this cell, as the frequency band in a given window is called, is ignored in the process. In this manner it is made extremely difficult to extract the encoded information from the signal without the use of the exact masks used in the encoding process. This is one place in which the stega-cipher process departs from traditional steganographic implementations, which offer a trivial decode opportunity if one knows the information is present. While this increases the information storage capacity of the carrier signal, it makes decoding trivial, and further degrades the signal. Note that it is possible and desirable to modify the boolean cell flag function so that it returns true < 50% of the time. In general, the fewer cells actually used in the encode, the more difficult they will be to find and the less degradation of content will be caused, provided the function is designed correctly. There is an obvious tradeoff in storage capacity for this increased security and quality.

The encoder proceeds in this manner until a complete copy of the additional information has been encoded in the carrier signal. It will be desirable to have the encoder encode multiple copies of the additional information continuously over the duration of the carrier signal, so that a complete instance of this information may be recovered from a smaller segment of a larger signal which has been split into discontinuous pieces or otherwise edited. It is therefore desirable to minimize the size of the information to be encoded using both compact design and pre-encoding compression, thus maximizing redundant encoding, and recoverability from smaller segments. In a practical implementation of this system it is likely the information will be first compressed by a known method, and then encrypted using public-key techniques, before being encoded into the carrier signal.

The encoder will also prepare the package of additional information so that it contains an easily recognizable start of message delimeter, which can be unique to each encoding and stored along with the keys, to serve as a synchronization signal to a decoder. The detection of this delimeter in a decoding window signifies that the decoder can be reasonably sure it is aligned to the sample stream correctly and can proceed in a methodic window by window manner. These delimeters will require a number of bits which minimizes the probability that this bit sequence is not reproduced in a random occurrence, causing an accidental misalignment of the decoder. A minimum of 256 bits is recommended. In the current implementation 1024 bits representing a start of message delimeter are used. If each sample is random, then each bit has a 50% probably of matching the delimeter and the conditional probability of a random match would be $1/2^{1024}$. In practice, the samples are probably somewhat less than random, increasing the probability of a match somewhat.

The decode process uses the same masks in the same manner, only in this case the information is extracted one bit at a time from the carrier signal.

The decoder is assumed to have access to the proper masks used to encode the information originally. These masks might be present in a database, which can be indexed by a value, or values computed from the original content, in a manner insensitive to the modifications to the content caused by the stega-cipher process. So, given an arbitrary piece of content, a decoder might first process the content to generate certain key values, and then retrieve the decode masks associated with the matching key values from the database. In the case where multiple matches occur, or none are found, it is conceivable that all mask sets in the database could be tried sequentially until a valid decode is achieved, or not, indicating no information is present.

In the application of this process, it is anticipated that encoding operations may be done on a given piece of content up to 3 times, each adding new information and using new masks, over a sub-segment of the content, and that decode operations will be done infrequently. It is anticipated that should it become necessary to do a search of a large number of masks to find a valid decode, that this process can be optimized using a guessing technique based on close key matching, and that it is not a time critical application, so it will be feasible to test large numbers of potential masks for validity on a given piece of content, even if such a process takes days or weeks on powerful computers to do a comprehensive search of known mask sets.

The decode process is slightly different in the following respect. Whereas the encoding process can start at any arbitrary point in the sample stream, the decode process does not know where the encode process began (the exact offset in samples to the start of the first window). Even though the encode process, by convention, starts with sample 0, there is no guarantee that the sample stream has not been edited since encoding, leaving a partial window at the start of the sample stream, and thus requiring the decoder to find the first complete window to start the decode. Therefore, the decode process will start at the first sample, and shift the sample window along by 1 sample, keeping the window index at 0, until it can find a valid decode delimeter encoded in the window. At this point, the decoder knows it has synchronized to the encoder, and can then proceed to process contiguous windows in a more expedient manner.

Example Calculations based on the described implementation for adding copyright certificate information to CD quality digital audio:

In a stream of samples, every 128 samples will contain, on average 64 bits of certificate related information. Digital audio is composed of 16 bit samples, at 44.1 Khz, or 44,100 samples per second. Stereo audio provides 2 streams of information at this rate, left and right, or 88,200 samples per second. That yields approximately 689 contiguous sample windows (of 128 samples) per second in which to encode information. Assume a song is 4 minutes long, or 240 seconds. This yields 240 * 689 = 165,360 windows, which on average (50% utilization) contain 64 bits (8 bytes) each of certificate information. This in turns gives approximately 1291Kb of information storage space per 4 minute stereo song (1.2 MB). There is ample room for redundant encoding of information continuously over the length of the content. Encoding 8 bytes for every 256 bytes represents 3.1% of the signal information. Assuming that a copyright certificate requires at most approximately 2048 bytes (2K), we can encode the same certificate in 645 distinct locations within the recording, or approximately every 37/100ths of a second.

Now to account for delimeters and synchronization information. Assuming a sync marker of 1024 bits to avoid random matches, then we could prefix each 2K certificate block with this 1024 bit marker. It takes 256 windows to store 2K, and under this proposed scheme, the first 16 windows are reserved for the sync marker. A decoder could search for this marker by progressively matching each of the first 16 windows (64 bits at a time) against the corresponding portion of the sync marker. The decoder could reset the match advancing through the sample stream, as soon as one window did not conform to the sync marker, and proceed in this manner until it matches 16 consecutive windows to the marker, at which point it is synchronized.

Under this scheme, 240 windows, or 1.92K remain for storing certificate information, which is not unreasonable.

## IV. Possible Problems, Attacks and Subsequent Defenses

### A. Randomization

The attacker simply randomizes the least significant bits of each data point in the transform buffer, obliterating the synchronization signal and the watermark. While this attack can remove the watermark, in the context in which stega-cipher is to be used, the problem of piracy is kept to a minimum at least equal to that afforded by traditional media, since the system will not allow an unwatermarked piece of content to be traded for profit and watermarks cannot be forged without the proper keys, which are computationally difficult to obtain by brute-force or cryptanalysis. In addition, if the encoding is managed in such a way as to maximize the level of changes to the sample stream to be just at the threshold below human perception, and the scheme is implemented to anticipate randomization attempts, it is possible to force the randomization level to exceed the level that can be perceived and create destructive artifacts in the signal, in much the same manner as a VHS cassette can be manufactured at a minimal signal level, so that a single copy results in unwatchable static.

### B. Low Bit-Depth Bitmaps (black & white images)

These bitmaps would be too sensitive to the steganization process, resulting in unacceptable signal degradation, and so are not good candidates for the stega-cipher process. The problem may be circumvented by inflating bit-depth, although this is an inefficient use of space and bandwidth.

### C. Non-Integer Transforms

The FFT is used to generate spectral energy information for a given audio signal. This information is not usually in integer format. Computers use methods of approximation in these cases to represent the real numbers (whole numbers plus fractional amounts). Depending on the exact value of the number to be represented slight errors, produced by rounding off the nearest real number that can be completely specified by the computer occur. This will produce some randomization in the least significant bit or bits. In other words, the same operation on the same sample window might yield slightly different transform values each time. It is possible to circumvent this problem using a modification to the simple LSB steganographic technique described later. Instead of looking at the LSB, the stega-cipher can use an energy quantization technique in place of the LSB method. Some variant of rounding the spectral energy values up or down, with a granularity greater than the rounding error should work, without significantly degrading the output samples.

## V. A Method and Protocol For Using the Stega-Cipher

The apparatus described in the claims below operates on a window by window basis over the sample stream. It has no knowledge of the nature of the specific message to be encoded. It merely indexes into a bit stream, and encodes as many of those bits as possible into a given sample window, using a map determined by the given masks.

The value of encoding information into a single window in the sample stream using such an apparatus may not be inherently apparent until one examines the manner in which such information will be used. The protocol discussed in this section details how messages which exceed the encoding capacity of a single sample window (128 samples) may be assembled from smaller pieces encoded in the individual windows and used to defend copyrights in an online situation.

An average of 64 bits can be encoded into each window, which equals only 8 bytes. Messages larger than 8 bytes can be encoded by simply dividing the messages up and encoding small portions into a string of consecutive windows in the sample stream. Since the keys determine exactly how many bits will be encoded per window, and an element of randomness is desirable, as opposed to perfect predictability, one cannot be certain exactly how many bits are encoded into each window.

The start of each message is marked by a special start of message delimeter, which, as discussed above is 1024 bits, or 128 bytes. Therefore, if precisely 8 bytes are encoded per window, the first 16 windows of any useable message in the system described here are reserved for the start of message delimeter. For the encoder, this scheme presents little challenge. It simply designates the first sample window in the stream to be window 0, and proceeds to encode the message delimeter, bit-by-bit into each consecutive window. As soon as it has processed the last bit of the SOM delimeter it continues by encoding 32 bits representing the size, in bytes of the complete message to follow. Once the 32nd and final bit of the size is encoded, the message itself is encoded into each consecutive window, one bit at a time. Some windows may contain more encoded bits then others, as dictated by the masks. As the encoder processes each window in the content it increments its window counter. It uses this counter to index into the window mask. If the number of windows required to encode a complete message is greater than the size of this mask, 256 bits in this case, or 256 windows, then it simply resets the counter after window

255, and so on, until a complete message is encoded. It can then start over, or start on a new message.

The decoder has a bigger challenge to face. The decoder is given a set of masks, just like encoder. Unlike the encoder, the decoder cannot be sure that the first series of 128 samples it receives are the window 0 start of message, encoded by the decoder. The sample stream originally produced by an encoder may have been edited by clipping its ends randomly or splicing pieces together. In that case, the particular copy of the message that was clipped is unrecoverable. The decoder has the start of message delimeter used to encode the message that the decoder is looking for. In the initial state, the decoder assumes the first window it gets is window 0. It then decodes the proper number of bits dictated by the masks it was given. It compares these bits to the corresponding bits of the start of message delimeter. If they match, the decoder assumes it is still aligned, increments the window counter and continues. If the bits do not match, the decoder knows it is not aligned. In this case, it shifts one more sample onto the end of the sample buffer, discarding the first sample, and starts over. The window counter is set to 0. The decoder searches one sample at a time for an alignment lock. The decoder proceeds in this manner until it has decoded a complete match to the start of message delimeter or it exhausts the sample stream without decoding a message. If the decoder can match completely the start of message delimeter bit sequence, it switches into aligned mode. The decoder will now advance through the sample stream a full window at a time (128 samples). It proceeds until it has the 32 bits specifying the message size. This generally won't occupy more than 1 complete window. When the decoder has locked onto the start of message delimeter and decoded the message size, it can now proceed to decode as many consecutive additional windows as necessary until it has decoded a complete message. Once it has decoded a complete message, the state of the decoder can be reset to un-synchronized and the entire process can be repeated starting with the next 128 sample window. In this manner it is not absolutely necessary that encoding windows be contiguous in the sample stream. The decoder is capable of handling random intervals between the end of one message and the start of another.

It is important to note that the circuit for encoding and decoding a sample window does not need to be aware of the nature of the message, or of any structure beyond the start of message delimeter and message size. It only needs to consider a single sample window, its own state (whether the decoder is misaligned, synchronizing, or synchronized) and what bits to encode/decode.

Given that the stega-cipher apparatus allows for the encoding and decoding of arbitrary messages in this manner, how can it be used to protect copyrights?

The most important aspect of the stega-cipher in this respect is that fact that it makes the message integral with the content, and difficult to remove. So it cannot be eliminated simply by removing certain information prepended or appended to the sample stream itself. In fact, removing an arbitrary chunk of samples will not generally defeat the stega-cipher either.

Given that some information can be thus integrated with the content itself, the question is then how best to take advantage of this arrangement in order to protect copyrights.

The following protocol details how the stega-cipher will be exploited to protect copyrights in the digital domain.

In a transaction involving the transfer of digitized content, there are at least 3 functions involved:

The Authority is a trusted arbitrator between the two other functions listed below, representing parties who actually engage in the transfer of the content. The Authority maintains a database containing information on the particular piece of content itself and who the two parties engaged in transferring the content are. The Authority can perform stega-cipher encoding and decoding on content.

The Publisher, or online distributor is the entity which is sending the copyrighted content to another party. The Publisher can perform stega-cipher encoding and decoding on content.

The Consumer is the person or entity receiving the copyrighted content, generally in exchange for some consideration such as money. The consumer cannot generally perform stega-cipher encoding or decoding on content.

Each of these parties can participate in a message exchange protocol using well known public-key cryptographic techniques. For instance, a system licensing RSA public key algorithms might be used for signed and encrypted message exchange. This means that each party maintains a public key / private key pair, and that the public keys of each party are freely available to any other party. Generally, the Authority communicates via electronic links directly only to the Publisher and the Consumer communicates directly only with the publisher.

Below is an example of how the protocol operates from the time a piece of content enters an electronic distribution system to the time it is delivered to a Consumer.

A copyright holder (an independent artist, music publisher, movie studio, etc.) wishes to retail a particular title online. For instance, Sire Records Company might wish to distribute the latest single from Seal, one of their musical artists, online. Sire delivers a master copy of this single, "Prayer for the Dying", to the Authority, Ethical Inc. Ethical converts the title into a format suitable for electronic distribution. This may involve digitizing an analog recording. The title has now become content in the context of this online distribution system. The title is not yet available to anyone except Ethical Inc., and has not yet been encoded with the stega-cipher watermark. Ethical generates a Title Identification and Authentication

(TIA) certificate. The certificate could be in any format. In this example it is a short text file, readable with a small word-processing program, which contains information identifying

the title
the artist
the copyright holder
the body to which royalties should be paid
general terms for publishers' distribution
any other information helpful in identifying this content

Ethical then signs the TIA with its own private key, and encrypts the TIA certificate plus its signature with its own public key. Thus, the Ethical can decrypt the TIA certificate at a later time and know that it generated the message and that the contents of the message have not been changed since generation.

Sire Records, which ultimately controls distribution of the content, communicates to the Ethical a specific online Publisher that is to have the right of distribution of this content. For instance, Joe's Online Emporium. The Authority, Ethical Inc. can transmit a short agreement, the Distribution Agreement to the Publisher, Joe's Online Emporium which lists

the content title
the publisher's identification
the terms of distribution
any consideration paid for the right to distribute the content
a brief statement of agreement with all terms listed above

The Publisher receives this agreement, and signs it using its private key. Thus, any party with access to the Joe's Online Emporium's public key could verify that the Joe's signed the agreement, and that the agreement has not been changed since

Joe's signed it. The Publisher transmits the signed Distribution Agreement to the Authority, Ethical Inc.

Ethical Inc. now combines the signed TIA certificate and the Distribution Agreement into a single message, and signs the entire message using its private key. Ethical has now created a Publisher Identification message to go into its own stega-cipher channel in the content. Ethical Inc. now generates new stega-cipher masks and encodes this message into a copy of the content using a stega-cipher encoder. The Authority saves the masks as a Receipt in a database, along with information on the details of the transfer, including the title, artist and publisher.

Ethical then transfers this watermarked copy to the Joe's Online Emporium, the Publisher. Well known encryption methods could be used to protect the transfer between the Authority and the Publisher. The Authority may now destroy its copy, which the Publisher has received. The Publisher, Joe's Online Emporium now assumes responsibility for any copies made to its version of the content, which is a Publisher Master copy.

Finally, the Consumer, John Q. Public wishes to purchase a copy of the content from Joe's Online Emporium. Joe's Emporium sends the John Q. Public a short agreement via an electronic communication link, similar to Publisher's Distribution Agreement, only this is a Purchase Agreement, which lists

the content title
consumer identification
the terms of distribution
the consideration pas for the content
a brief statement of agreement with the terms above

John Q. Public signs this agreement with his private key and returns it to the Joe's Online Emporium. The Publisher, Joe's prepares to encode its own stega-cipher watermark onto a copy of the content by generating a set of masks for the algorithm. Joe's Online Emporium then stores these masks (a receipt) in its own database, indexed by title and consumer. Joe's Online Emporium signs the agreement received from John Q. Public with the Emporium's own private key, and forwards it to the Authority, Ethical Inc., along with a copy of the masks. It is important to note that this communication should be done over a secured channel. The Authority verifies the Publisher and Consumer information and adds its own signature to the end of the message, approving the transaction, creating a Contract of Sale. The Authority adds the Publisher's receipt (mask set) to its database, indexed by the title, the publisher, and the consumer identification. The Authority signs the Contract of Sale by encrypting it with their private key. So anyone with the Authority's public key (any Publisher) could decrypt the Contract of Sale and verify it, once it was extracted from the content. The Publisher then transmits the signed Contract of Sale back to the Publisher, who uses a stega-cipher device to imprint this Contract as its own watermark over the content. The Publisher then transmits the newly watermarked copy to the Consumer, who is accepting responsibility for it. The Publisher destroys their version of the consumer's copy.

If this procedure is followed for all content distribution within such an online system then it should be possible for the Authority to identify the owner of a piece of content which appears to be unauthorized. The Authority could simply try its database of stega-cipher keys to decode the watermark in the content in question. For instance, if a copy of Seal's latest single originally distributed with stega-cipher watermarks showed up on an Internet ftp site the Authority should be able to extract a TIA Certificate and Distribution Agreement or a Contract of Sale identifying the responsible party. If a Publisher sold this particular copy to a Consumer, that particular publisher should be able to extract a Contract of Sale, which places responsibility with the Consumer. This is not a time critical application, so even if it takes days or weeks, it is still worthwhile.

In a modification to the protocol discussed above, each Publisher might act as its own Authority. However, in the context of online services, this could open avenues of fraud committed by the collusion of certain Publishers and Consumers. Using an Authority, or one of several available Authorities to keep records of Publisher-Consumer transactions and verify their details decreases the likelihood of such events.

It should also be obvious that a similar watermarking system could be used by an individual entity to watermark its own content for its own purposes, wether online or in physical media. For instance, a CD manufacturer could incorporate unique stega-cipher watermarks into specific batches of its compact discs to identify the source of a pirate ring, or to identify unauthorized digital copies made from its discs. This is possible because the stega-cipher encoding works with the existing formats of digital samples and does not add any new structures to the sample data that cannot be handled on electronic or mechanical systems which predate the stega-cipher.

## VI. Increasing Confidence in the Stega-Cipher

The addition of a special pre-encoding process can make stega-cipher certificates even more secure and undeniable. Hash values may be incorporated which match exactly the content containing the watermark to the message in the watermark itself. This allows us a verification that the watermark decoded was encoded by whomever signed it into this precise location in this specific content.

Suppose one wants to use a 256 bit (32 byte) hash value which is calculated with a secure one-way hash function over each sample in each sample window that will contain the message. The hash starts with a seed value, and each sample that would be processed by the encoder when encoding the message is incorporated into the hash as it is processed. The result is a 256 bit number one can be highly confident is unique, or sufficiently rare to make intentionally duplicating it with another series of samples difficult.

It is important that the hash function be insensitive to any changes in the samples induced by the stega-cipher itself. For instance, one might ignore the least significant bit of each sample when computing the hash function, if the stega-cipher was implemented using a least significant bit encode mode.

Based on the size of the non-hash message, one knows the hash-inclusive message requires 32 more bytes of space. One can now calculate the size of a signed encrypted copy of this message by signing and encrypting exactly as many random bytes as are in the message, and measuring the size of the output in bytes. One now knows the size of the message to be encoded. One can pre-process the sample stream as follows.

Proceed through the stega-cipher encode loop as described in the claims. Instead of encoding, however, calculate hash values for each window series which will contain the message, as each sample is processed. At the end of each instance of "encoding" take the resultant hash value and use it to create a unique copy of the message which includes the hash value particular to the series of sample windows that will be used to encode the message. Sign and encrypt this copy of the message, and save it for encoding in the same place in the sample stream.

A memory efficient version of this scheme could keep on hand the un-hashed message, and as it creates each new copy, back up in the sample stream to the first window in the series and actually encode each message, disposing of it afterwards.

The important result is evident on decoding. The decoding party can calculate the same hash used to encode the message for themselves, but on the encoded samples. If the value calculated by the decoding party does not match the value contained in the signed message, the decoder is alerted to the fact that this watermark was transplanted from somewhere else. This is possible only with a hash function which ignores the changes made by the stega-cipher after the hash in the watermark was generated.

This scheme makes it impossible to transplant watermarks, even with the keys to the stega-cipher.

## Appendix - Psuedo-code

```
const int WINDOW_RESET = 256;
const int WINDOW_SIZE = 128;
const int MARKER_BITS = 1024;
const int CHUNK_BITS = 2048 * 8;

int window_offset;
int msg_bit_offset;
int frequency_offset;
Boolean useCell;

/* 8 bits per bye, 1 byte per char */
unsigned char frequency_mask[WINDOW_SIZE/8];
unsigned char window_mask[WINDOW_RESET/8];
unsigned char msg_start_marker[MARKER_BITS/8];
unsigned char msg_end_marker[MARKER_BITS/8];
Int16 amplitude_sample_buffer[WINDOW_SIZE];
float power_frequency_buffer[WINDOW_SIZE];
unsigned char message_buffer[CHUNK_BITS/8];

void doFFT(Int16 *amp_sample_buffer, float *power_freq_buffer,int size);
void doInverseFFT(Int16 *amp_sample_buffer, float *power_freq_buffer,int size);
void initialize();
Bit getBit(unsigned char *buffer,int bitOffset);
Boolean map(Bit window_bit, Bit band_bit, int window, int frequency);
Boolean getSamples(Int16 *amplitude_sample_buffer,int samples);
void encode()

void initialize()
{
        /* message to be encoded is generated */
        /* message is prefixed with 1024 bit msg_start_marker */
        /* message is suffixed with 1024 bit msg_end _marker */
        /* remaining space at end of message buffer padded with random bits */
        window_offset = 0;
        msg_bit_offset = 0;
        frequency_offset = 0;
        frequency_mask loaded
        window_mask loaded
        zeroAmpSampleBuffer();
}
```

```
Boolean getSamples(Int16 *buffer,int samples)
{
        /* get samples number of samples and shift them contiguously into the sample
           buffer from right to left*/
        if(samples < samples available)
                return false;
        else
                return true;
}

void doFFT(Int16 *sample_buffer, float *spectrum_buffer, int size)
{
        calculate FFT on sample_buffer, for size samples
        store result in spectrum buffer
}

void doInverseFFT(Int16 *sample_buffer,float *spectrum_buffer,int size)
{
        calculate inverse FFT on spectrum_buffer
        store result in sampe_buffer
}

Bit getBit(unsigned char *buffer,in bitOffset)
{
        returns value of specified bit in specified buffer
        either 0 or 1, could use Boolean (true/false) values for bit set of bit off
}

Boolean map(Bit window_bit,Bit band_bit,int window, int frequency_
{
        /* this is the function that makes the information difficult to find */
        /* the inputs window_bit and band_bit depend only on the mask values
                used for encoding the information, they are 1) random, 2) secret */
        /* window and frequency values are used add time and frequency band dependent
                complexity to this function */
        /* this function is equivalent to a Boolean truth table with window * frequency * 4
        possible input combinations and 2 possible output */
        /* for any input combination, the output is either true or false */
        /* window ranges from 0 to WINDOW_RESET -1 */
        /* frequency ranges from 0 to WINDOW_SIZE - 1 */
        return calculated truth value
}
```

```
    void encodeBit(float *spectrum_buffer,int freq_offset,Bit theBit)
    {
            /* modifies the value of the cell in spectrum_buffer, indexed by freq_offset
                    in a manner that distinguishes each of the 2 possible values of theBit,
                    1 or 0
            */
            /* suggested method of setting the Least Significant bit of the cell == theBit */
            /* alternative method of rounding the value of the cell upward or downward to
                    certain fractional values proposed
                    i.e. <= .5 fractional remainder signifies 0, > .5 fraction remainder
                            signifies 1
            */
    }

void encode()
{
        initialize();


        do {


        if(getSamples(amplitude_sample_buffer) == false)
                return


        doFFT(amplitude_sample_buffer,power_frequency_buffer,WINDOW_SIZE);

        for (frequency_offset = 0; frequency_offset < WINDOW_SIZE;
        frequency_offset++){


                useCell = map(getBit(window_mask,window_offset),
                                getBit(frequency_mask,frequency_offset),
                                window_offset, frequency_offset);

                if(useCell == true){
                        encodeBit(power_frequency_buffer,frequency_offset,
                                getBit(message_buffer,msg_bit_offset));
                        message_bit_offset ++;
                        if(msg_bit_offset == MESSAGEBITS){
                                initialize();
                                break; /* exit frequency loop */
                        }
                }
        }
```

```
            doInverseFFT(amplitude_sample_buffer,power_frequency_buffer,
                    WINDOW_SIZE);

            outputSamples(amplitude_sample_buffer);

            window_offset++;
            if(window_offset == WINDOW_RESET){
                    window_offset = 0;
            }


        }while(true);
}
```

The encode() procedure processes an input sample stream using the specified frequency and window masks as well as a pre-formatted message to encode.

encode() processes the sample stream in windows of WINDOW_SIZE samples, contiguously distributed in the sample stream, so it advances WINDOW_SIZE samples at a time.

For each sample window, encode() first compute the FFT of the window, yielding its Power Spectrum Estimation. For each of these window PSEs, encode() then uses the map() function to determine where in each PSE to encode the bits of the message, which it reads from the message buffer, on ebit at a time. Each time map() returns true, encode() consumes another sample from the message.

After each window is encoded, encode() computes the inverse FFT on the PSE to generate a modified sample window, which is then output as the modified signal. It is important the sample windows NOT overlap in the sample stream, since this would potentially damage the preceeding encoding windows in the stream.

Once the message is entirely encoded, including its special end of message marker bit stream, encode() resets it internal variables to begin encoding the message once more in the next window. encode() proceeds in this manner until the input sample stream is exhausted.

```
enum {
        Synchronizing,
        Locked
}; /* decode states */
```

```
unsigned char message_end_buffer[MARKER_BITS];

Bit decodeBit(float *spectrum_buffer,int freq_offset)
{
        /* reads the value of the cell in spectrum_buffer, indexed by freq_offset
                in a manner that distinguishes each of the 2 possible values of an
                encoded bit, 1 or 0
        */
        /* suggested method of testing the Least Significant bit of the cell */
        /* alternative method of checking the value of the cell versus certain fractional
remainders proposed.
                i.e. <= .5 fractional remainder signifies 0, > .5 fraction remainder
                        signifies 1
        */
        return either 1 or 0 as appropriate
}

Boolean decode()
{
        /* Initialization */
        state = Synchronizing
        window_offset = 0;
        set frequency mask
        set window mask
        clear sample buffer
        int nextSamples = 1;
        int msg_start_offset = 0;
        clear message_end_buffer
        Bit aBit;
        Boolean bitsEqual;

        do {

                if(state == Synchronizing) {
                        nextSamples = 1;
                        window_offset = 0;
                }
                else
                        nextSamples = WINDOW_SIZE;

                if(getSamples(amplitude_sample_buffer) == false)
                        return false;
```

```
                  doFFT(amplitude_sample_buffer,power_frequency_buffer,
                           WINDOW_SIZE);  /* 2 */

                  for (frequency_offset = 0; frequency_offset < WINDOW_SIZE;
                  frequency_offset++){


                         useCell = map(getBit(window_mask,window_offset),
                                  getBit(frequency_mask,frequency_offset),
                                  window_offset, frequency_offset);

                        if(useCell == true){
                                 aBit = decodeBit(power_frequency_buffer,
                                                   frequency_offset);
                                 setBit(message_buffer,message_bit_offset,aBit);
                                 message_bit_offset ++;
                        }
                        else
                                 continue;
                       if(state == Synchronizing){
                                bitsEqual =
                                compareBits(message_start_marker,message_buffer,
                                          message_bit_offset);
                               if(!bitsEqual){
                                          message_bit_offset = 0;
                                          misaligned = true;
                                          break; /* exit frequency loop */
                               }
                              else if (message_bit_offset == MARKER_BITS)
                                          state == Locked;
                      }
                      else {
                               /* locked onto encoded stream */
                               shift aBit into right side of message_end_buffer
                               bitsEqual = compareBits(message_end_buffer,
                                         msg_end_marker,MARKER_BITS);
                              if(bitsEqual)
                                          return true;
                      }
             }

     } while (true);
}
```

The decode() procedure scans an input sample stream using specified window and frequency masks, until it either decodes a valid message block, storing it in a message buffer, or exhausts the sample stream.

The decode() procedure starts in state Synchronizing, in which it does not know where in the sample stream the encoding windows are aligned. The procedure advances the sample window through the sample stream one sample at a time, performing the FFT calculation on each window, and attempting to decode valid message bits from the window. As it extracts each bit using the map() function, the decode() procdure compares these bits against the start of message marker. As soon as a mismatch is detected, the decode() procedure knows it is not yet properly aligned to an encoding window, and immediately ceases decoding bits from the current window and moves to the next window, offset by 1 sample. The decode() procedure continues in this manner until it matches successfully the complete bitstream of a start of message marker. At this point the decode() procedure assumes it is aligned to an encoded message and can then decode bits to the message buffer quickly, advancing the sample window fully at each iterations. It is now in Locked mode. For each bit it stores in the message buffer when in Locked mode, the decode() procedure also shifts the same bit value into the least significant bit of the message_end_buffer. After each bit is decoded in Locked mode, the decode() procedure checks compares the message_end_buffer with the msg_end_marker in a bit by bit manner. When a complete match is found, decode() is finished and returns true. If the sample stream is exhausted before this occurs, decode() returns false. If decode() returns true, a valid message is stored in the message buffer, including the start and end of message markers.

**Claims**

1. A steganographic method comprising the steps of :

using random keys in combination with steganography to encode additional information into digitized samples such that a signal generated from the modified sample stream is not significantly degraded and such that the additional information cannot be extracted without the keys and such that the signal generated from the modified sample stream will be degraded by attempts to erase, scramble, or otherwise obliterate the encoded additional information.

2. An apparatus for encoding or decoding a message, represented as series of data bits into or out of a series of digitized samples, comprising:

a) a sample buffer for holding and accessing and transforming digitized samples;

b) a digital signal processor capable of performing fast fourier transforms;

c) a memory to contain information representing

1) primary mask,
2) convolutional mask,
3) start to message delimiter,
4) a mask calculation buffer,
5) a message buffer,
6) an integer representing a message bit index,
7) a position integer M representing message size,
8) an integer representing an index into said primary mask,
9) an integer representing an index into said convolution mask,
10) an integer representing the state of a decode process,
11) a table representing a map function;
12) a flag indicating a complete message has been decoded or encoded,

13) a positive integer S representing a number of samples to read into said sample buffer, and

14) a flag indicating the size of a message which has been decoded;

d) an input to acquire digital samples;

e) an output to output modified digital samples;

f) an input for inputting the values of (c1) - (c5) and (c11) and (c13);

g) an output to output the message stored in (c5) as the result of a decode process and the value of (c10) to an attached digital circuit;

h) at least one data bus to transfer information from

(d) to (a),

(a) to (b),

(b) to (a),

(a) to (e),

(f) to (c), and

(c) to (e); and

i) a clock which generates a clock signal to drive (b) and control the operation of the apparatus.

3. A method of encoding information into a sample stream of data, said method comprising the steps of:

A) generating a mask set to be used for encoding, said set including:

a random or pseudo-random primary mask,

a random or pseudo-random convolution mask,

a random or pseudo-random start of message delimiter, wherein said mask set can be concatenated and manipulated as a single bit stream;

B) obtaining a message to be encoded;

C) generating a message bit steam to be encoded such that the stream includes

1) a start of message delimiter, and

2) an integer representing the number of message bytes to follow the message;

D) loading the message bit stream, a map table, the primary mask, the convolution mask, and the start of message delimiter into a memory;

E) resetting a primary mask index, a convolution mask and message bit index, and setting the message size integer equal to the total number of bits in the message bit stream;

F) clearing a message encoded flag;

G) reading a window of samples from a sample input device and storing them sequentially in a sample buffer;

H) resetting the primary mask index and looping through the sample buffer from a first sample to a last sample incrementing the primary mask index each time a sample is visited, such that for each sample position, a value of the mapping function is computed, which is either true or false, by using a bit of the primary mask representing a current sample and a bit of the convolution mask indicated by the convolution index to calculate an offset in the map table;

I) obtaining the bit value stored in the map table and encoding the bit of the message indicated by the message bit index into the current sample if the bit value obtained from the map table is a certain value and incrementing the message bit index, determining whether the message bit index equals the number of message bits, and if it does re-performing step A), setting the message encoded flag, and exiting the loop;

J) outputting the modified samples in the sample buffer, and if the message encoded flag is set jumping back to said step E);

K) incrementing the convolution index, wherein if the convolution index equals the length of the convolution mask in bits then set the convolution index to 0; and

L) jumping back to step G).

4. A method of encoding information into a sample stream of data, comprising the steps of:

A) generating a mask set to be used for encoding, including:
a random or pseudo-random primary mask,
a random or pseudo-random convolution mask, and
a random or pseudo-random start of message delimiter, wherein said mask set can be concatenated and manipulated as a single bit stream;

B) inputting a message to be encoded;

C) generating a message bit stream to be encoded including
a start of message delimiter, and
an integer representing of number of message bytes to follow the message;

D) loading the message bit stream, a map table, and the mask set into a memory;

E) resetting a primary mask index, a convolution mask and message bit index, setting the message size index equal to the number of bits in the message bitstream, and clearing a message encoded flag;

F) reading a window of samples of the inputted message and storing the samples sequentially in a sample buffer;

G) computing a spectral transform of the samples in the buffer;

H) obtaining the bit value stored in the map table, wherein if the bit value is true, then encoding the bit of the message indicated by the message bit index into the current sample and incrementing the message bit index, where the message bit index equals the number of message bits, and then reperforming step A), setting the message encoded flag, and exiting the loop;

I) computing the inverse spectral of the spectral values stored in the sample buffer,

J) outputting the values in the sample buffer, and if the sample encoded flag is set, then clear the flag and jump back to step E);

K) incrementing the convolution index and when the convolution index equals the length of the convolution mask in bits resetting the convolution index; and

L) jumping back to step F).

5. The method of claim 3 wherein the encoding of the message bit into the sample in step I includes encoding a single bit of the sample to match the message bit.

6. The method of claim 4 wherein the encoding of the message bit into the sample in step H includes altering the sample value such that said sample value falls within a prespecified range of valves relative to its original value.

7. A method of decoding information from a sample stream of data, comprising the steps of:

A) obtaining a mask set including:

(1) a random or pseudo-random primary mask,

(2) a random or pseudo-random convolution mask, and

(3) a random or pseudo-random start of message delimiter;

B) loading a map table, and the mask set into a memory,

C) resetting a primary mask index and convolution mask index and setting a message size integer equal to 0;

D) clearing a message decoded flag;

E) setting a state of the decode process to SYNCHRONIZED;

F) checking the state of the decode process and if the decode state is UNSYNCHRONIZED, setting a number of samples to equal 1 and resetting the convolution index to 0; otherwise, setting the number of samples to equal S $(S \geq 1)$;

G) reading the number of samples specified in step F) into a sample buffer;

H) resetting the primary mask index, and looping through the sample buffer from the first sample to the last sample, incrementing the primary mask index each time, and for each sample position, computing the value of a mapping function to calculate an offset into the map table;

I) obtaining the bit value in the map table, and if the value is true, decoding the bit of the message indicated by the message bit index, storing the bit into the message buffer at the message bit index, and incrementing the message bit index;

J) comparing the decoded bits in the message buffer to the start of message delimiter, and if the number of bits in the message buffer is less than or equal to the number of bits in the start of message delimiter and the bits match, then setting the state of the decode process to SYNCHRONIZED; otherwise setting the state of the decode process to UNSYNCHRONIZED;

K) if the state of the decode process is SYNCHRONIZED and the number of bits in the message buffer is greater than or equal to the sum of the number of bits of the start of delimiter and the message size, then setting the state of the decode process to SYNC-AND-SIZE and copying certain bits from the message buffer to a message size integer container;

L) if the state of the decode process is SYNC-AND-SIZE and the number of bits in the message buffer divided by 8 is greater than or equal to the message size, then setting the message decoded flag, outputting the message and the message decoded flag and ending the method;

M) incrementing the convolution index, and if the convolution index equals the number of bits in the convolution mask resetting the convolution index; and

N) jumping to step F).

8. A method of decoding information from sampled data, comprising the steps of:

A) Obtaining a mask set including

(1) a random or pseudo-random primary mask,

(2) a random or pseudo-random convolution mask, and

(3) a random or pseudo-random start of message delimiter,

B) loading a map table, and the mask set into a memory;

C) resetting a primary mask index and convolution mask index and setting a message size integer equal to 0;

D) clearing a message decoded flag;

E) setting a state of the decode process to SYNCHRONIZED;

F) checking the state of the decode process and if the decode state is UNSYNCHRONIZED, setting a number of samples to equal 1 and resetting the convolution index to 0; otherwise, setting the number of samples to equal S (S>1);

G) reading the number of samples specified in step F) into a sample buffer;

H) computing a spectral transform of the samples stored in the sample buffer;

I) resetting the primary mask index and looping through the sample buffer from the first sample to the last sample, incrementing the primary mask index each time, and for each sample position, computing the value of a mapping function by using the bit of the primary mask corresponding to the primary mask index and the bit of the convolution masks indicated by the convolution phase to calculate an offset into the map table representing the mapping function;

J) obtaining a bit value stored in the map, and if the value is true, decoding the bit of the message indicated by the message bit index from the current sample, storing the bit into the message buffer at the message bit index, and incrementing the message bit index;

K) comparing the decoded bits in the message buffer to the start of message delimiter, and if the number of bits in the message buffer is less than or equal to the number of bits in the start of message delimiter and the bits match, then setting the state of the decode process to SYNCHRONIZED; otherwise, setting the state of the decode process UNSYNCHRONIZED;

L) if the state of the decode process is SYNCHRONIZED, and the number of bits in the message buffer is greater than or equal to the sum of the number of bits of the start of delimiter and the message size, then setting the state of the decode process to SYNC-AND-SIZE and copying certain bits from the message buffer to a message size integer container;

M) if the state of the decode process is SYNC-AND-SIZE and the number of bits in the message buffer divided by 8 is greater than or equal to the message size, then setting the message decoded flag, outputting the message and the message decoded flag and ending the method;

N) incrementing the convolution index, wherein if the convolution index equals the number of bits in the convolution mask, then resetting the convolution index; and

O) jumping to step F).

9. The method of claim 7 wherein the decoding of the message bit from the sample in step I includes reading a single bit of the sample.

10. The method of claim 7 wherein the decoding of the message bit from the sample in step I includes mapping a range of sample values onto a particular message bit value.

11. The method of claim 4 wherein the map table is defined such that any index of the map table directs the process to encode information.

12. The method of claim 1 wherein the samples are obtained from a sample stream representing digitized sound or music.

13. The method of claim 12 wherein the identical encode process is performed on two sample streams representing channel A and channel B of digitized stereo sound.

14. The method of claim 12 wherein the sample streams represent channel A and channel B of digitized stereo sound and are interleaved before being input as a single sample stream and are separated into two channels upon output.

15. The method of claim 1 wherein the samples are obtained from a sample stream representing digitized video.

16. The method of claim 1 wherein the samples are obtained from a sample stream representing a digitized image.

17. The apparatus of claim 2, further comprising a tamper-resistant packaging, enclosing said apparatus wherein circuitry and information stored therein are destroyed if said packaging is opened.

18. The method of claim 3, further comprising a pre-encoding step which customizes the message to be encoded including: calculating over which windows in the samples stream a message will be encoded, computing a secure one way hash function of the samples in those windows, and placing the resulting hash values in the message before the message is encoded;

wherein the hash calculating step includes: calculating the size of the original message plus the size of an added hash value, and pre-processing the sample stream for the purpose of calculating hash values of each series of windows that will be used to encode the message and creating a modified copy of the message containing the hash value such that each message containing a hash value matches each window series uniquely.

19. The method of claim 1, wherein an authority for on line distribution of content encodes at least one of the following items into a sample stream ;

the title,
the artist,
the copyright holder,
the body to which royalties should be paid, and
general terms for publisher distribution.

20. The method of claim 19, wherein the authority combines at least one item with a secure private key signed message from a publisher containing at least one of the following pieces of information:

the title,
the publisher's identification,
the terms of distribution,
any consideration paid for the right to distribute the content,
a brief statement of agreement, and

the publisher signs and encrypts the combined message using a public key cryptosystem and encodes the signed and encrypted message into the sample stream.

21. The method of claim 20, wherein a publisher obtains the encoded sample stream and additionally obtains information form the authority and combines this with a message received from a consumer, which has been signed using a public key cryptosystem and wherein the signed message contains at least one of the following data

the content title,
consumer identification,
the terms of distribution,
the consideration paid for the content,
a brief statement of agreement, and

the publisher uses a public key cryptosystem to sign the combined information and finally encodes the signed information.

22. The method of claim 1, wherein the sample stream is obtained from at least one audio track contained within a digitized movie, video game software, or other software.

23. The method of claim 1, wherein the sample stream is obtained from at least one digitized movie or still image contained within a video game or other software.

24. The method of claim 1, wherein encoded information is contained in the differences or relationship between samples or groups of samples.

25. The method of claim 4, wherein the encoding of the message bit into the sample in step H includes encoding a single bit of the sample to match the message bit.

26. The method of claim 3, wherein the encoding of the message bit into the sample in step I includes altering the sample value such that said sample value falls within a prespecified range of valves relative to its original value.

27. The method of claim 8, wherein the decoding of the message bit in step J includes reading a single bit of the sample.

28. The method of claim 8, wherein the decoding of the message bit in step J includes mapping a range of supply values onto a particular message bit value.

29. The method of claim 3, wherein the map table is defined such that any index of the map table directs the process to encode information.

30. The method of claim 7, wherein the map table is defined such that any index of the map table directs the process to encode information.

31. The method of claim 8, wherein the map table is defined such that any index of the map table directs the process to encode information.

32. The method of claim 4, further comprising a pre-encoding step which customizes the message to be encoded including: calculating over which windows in the samples stream a message will be encoded, computing a secure one way hash function of the samples in those windows, and placing the resulting hash values in the message before the message is encoded;

wherein the hash calculating step includes: calculating the size of the original message plus the size of an added hash value, and pre-processing the sample stream for the purpose of calculating hash values of each series of windows that will be used to encode the message and creating a modified copy of the message containing the hash value such that each message containing a hash value matches each window series uniquely.--

# PCT

## REQUEST

The undersigned requests that the present international application be processed according to the Patent Cooperation Treaty.

For receiving Office use only

International Application No.

International Filing Date

Name of receiving Office and "PCT International Application"

Applicant's or agent's file reference (if desired) (12 characters maximum) 2377/13

Box No. I TITLE OF INVENTION

STEGANOGRAPHIC METHOD AND DEVICE

Box No. II APPLICANT

Name and address: (Family name followed by given name; for a legal entity, full official designation. The address must include postal code and name of country.)

THE DICE COMPANY
P.O. Box 60471
Palo Alto, California 94306-0471

☐ This person is also inventor.

Telephone No.
(415) 326-4364

Facsimile No.

Teleprinter No.

State (i.e. country) of nationality:
US

State (i.e. country) of residence:
US

This person is applicant for the purposes of: ☒ all designated States ☐ all designated States except the United States of America ☐ the United States of America only ☐ the States indicated in the Supplemental Box

Box No. III FURTHER APPLICANT(S) AND/OR (FURTHER) INVENTOR(S)

Name and address: (Family name followed by given name; for a legal entity, full official designation. The address must include postal code and name of country.)

COOPERMAN, Marc S.
2929 Ramona
Palo Alto, California 94306
US

This person is:
☐ applicant only
☐ applicant and inventor
☒ inventor only (If this check-box is marked, do not fill in below.)

State (i.e. country) of nationality:

State (i.e. country) of residence:

This person is applicant for the purposes of: ☐ all designated States ☐ all designated States except the United States of America ☐ the United States of America only ☐ the States indicated in the Supplemental Box

☐ Further applicants and/or (further) inventors are indicated on a continuation sheet.

Box No. IV AGENT OR COMMON REPRESENTATIVE; OR ADDRESS FOR CORRESPONDENCE

The person identified below is hereby/has been appointed to act on behalf of the applicant(s) before the competent International Authorities as: ☒ agent ☐ common representative

Name and address: (Family name followed by given name; for a legal entity, full official designation. The address must include postal code and name of country.)

ALTMILLER, John C.
PIETRANTONIO, Frank
KENYON & KENYON
1025 Connecticut Ave., N.W.
Washington, DC 20036
US

Telephone No.
(202) 429-1776

Facsimile No.
(202) 429-0796

Teleprinter No.

☐ Mark this check-box where no agent or common representative is/has been appointed and the space above is used instead to indicate a special address to which correspondence should be sent.

Form PCT/RO/101 (first sheet) (1 July 1994; reprint January 1996)

Sheet No. -2-

**Continuation of Box No. III FURTHER APPLICANTS AND/OR (FURTHER) INVENTORS**

*If none of the following sub-boxes is used, this sheet is not to be included in the request.*

Name and address: *(Family name followed by given name; for a legal entity, full official designation. The address must include postal code and name of country.)*

MOSKOWITZ, Scott A.
20191 East Country Club Drive
Townhouse 4
North Miami Beach, Florida 33180
US

This person is:
☐ applicant only
☐ applicant and inventor
☒ inventor only *(If this check-box is marked, do not fill in below.)*

State (*i.e.* country) of nationality:

State (*i.e.* country) of residence:

This person is applicant for the purposes of: ☐ all designated States ☐ all designated States except the United States of America ☐ the United States of America only ☐ the States indicated in the Supplemental Box

Name and address: *(Family name followed by given name; for a legal entity, full official designation. The address must include postal code and name of country.)*

This person is:
☐ applicant only
☐ applicant and inventor
☐ inventor only *(If this check-box is marked, do not fill in below.)*

State (*i.e.* country) of nationality:

State (*i.e.* country) of residence:

This person is applicant for the purposes of: ☐ all designated States ☐ all designated States except the United States of America ☐ the United States of America only ☐ the States indicated in the Supplemental Box

Name and address: *(Family name followed by given name; for a legal entity, full official designation. The address must include postal code and name of country.)*

This person is:
☐ applicant only
☐ applicant and inventor
☐ inventor only *(If this check-box is marked, do not fill in below.)*

State (*i.e.* country) of nationality:

State (*i.e.* country) of residence:

This person is applicant for the purposes of: ☐ all designated States ☐ all designated States except the United States of America ☐ the United States of America only ☐ the States indicated in the Supplemental Box

Name and address: *(Family name followed by given name; for a legal entity, full official designation. The address must include postal code and name of country.)*

This person is:
☐ applicant only
☐ applicant and inventor
☐ inventor only *(If this check-box is marked, do not fill in below.)*

State (*i.e.* country) of nationality:

State (*i.e.* country) of residence:

This person is applicant for the purposes of: ☐ all designated States ☐ all designated States except the United States of America ☐ the United States of America only ☐ the States indicated in the Supplemental Box

Sheet No. -3-

**Box No.V** **DESIGNATION OF STATES**

The following designations are hereby made under Rule 4.9(a) *(mark the applicable check-boxes; at least one must be marked):*

**Regional Patent**

☐ **AP** **ARIPO Patent:** **KE** Kenya, **LS** Lesotho, **MW** Malawi, **SD** Sudan, **SZ** Swaziland, **UG** Uganda, and any other State which is a Contracting State of the Harare Protocol and of the PCT

☐ **EA** **Eurasian Patent:** **AZ** Azerbaijan, **BY** Belarus, **KZ** Kazakstan, **RU** Russian Federation, **TJ** Tajikistan, **TM** Turkmenistan, and any other State which is a Contracting State of the Eurasian Patent Convention and of the PCT

☒ **EP** **European Patent:** **AT** Austria, **BE** Belgium, **CH and LI** Switzerland and Liechtenstein, **DE** Germany, **DK** Denmark, **ES** Spain, **FR** France, **GB** United Kingdom, **GR** Greece, **IE** Ireland, **IT** Italy, **LU** Luxembourg, **MC** Monaco, **NL** Netherlands, **PT** Portugal, **SE** Sweden, and any other State which is a Contracting State of the European Patent Convention and of the PCT Finland

☐ **OA** **OAPI Patent:** **BF** Burkina Faso, **BJ** Benin, **CF** Central African Republic, **CG** Congo, **CI** Côte d'Ivoire, **CM** Cameroon, **GA** Gabon, **GN** Guinea, **ML** Mali, **MR** Mauritania, **NE** Niger, **SN** Senegal, **TD** Chad, **TG** Togo, and any other State which is a member State of OAPI and a Contracting State of the PCT *(if other kind of protection or treatment desired, specify on dotted line)* ........................................

**National Patent** *(if other kind of protection or treatment desired, specify on dotted line):*

☐ **AL** Albania
☐ **AM** Armenia ........................
☐ **AT** Austria ........................
☐ **AU** Australia ........................
☐ **AZ** Azerbaijan
☐ **BB** Barbados
☐ **BG** Bulgaria ........................
☐ **BR** Brazil ........................
☐ **BY** Belarus
☒ **CA** Canada
☐ **CH and LI** Switzerland and Liechtenstein
☒ **CN** China ........................
☐ **CZ** Czech Republic ........................
☐ **DE** Germany ........................
☐ **DK** Denmark ........................
☐ **EE** Estonia ........................
☐ **ES** Spain ........................
☒ **FI** Finland .(EP.)........................
☐ **GB** United Kingdom
☐ **GE** Georgia ........................
☐ **HU** Hungary ........................
☐ **IS** Iceland
☒ **JP** Japan ........................
☐ **KE** Kenya ........................
☐ **KG** Kyrgyzstan ........................
☐ **KP** Democratic People's Republic of Korea ........................
☒ **KR** Republic of Korea ........................
☐ **KZ** Kazakstan ........................
☐ **LK** Sri Lanka
☐ **LR** Liberia
☐ **LS** Lesotho ........................
☐ **LT** Lithuania
☐ **LU** Luxembourg ........................
☐ **LV** Latvia
☐ **MD** Republic of Moldova ........................
☐ **MG** Madagascar ........................
☐ **MK** The former Yugoslav Republic of Macedonia ........................
☐ **MN** Mongolia
☐ **MW** Malawi ........................
☐ **MX** Mexico ........................
☐ **NO** Norway
☐ **NZ** New Zealand ........................
☐ **PL** Poland ........................
☐ **PT** Portugal ........................
☐ **RO** Romania
☐ **RU** Russian Federation ........................
☐ **SD** Sudan
☐ **SE** Sweden
☒ **SG** Singapore
☐ **SI** Slovenia ........................
☐ **SK** Slovakia ........................
☐ **TJ** Tajikistan ........................
☐ **TM** Turkmenistan ........................
☐ **TR** Turkey ........................
☐ **TT** Trinidad and Tobago ........................
☐ **UA** Ukraine ........................
☐ **UG** Uganda ........................
☐ **US** United States of America ........................
☐ **UZ** Uzbekistan ........................
☐ **VN** Viet Nam ........................

Check-boxes reserved for designating States (for the purposes of a national patent) which have become party to the PCT after issuance of this sheet:

☐ ........................
☐ ........................
☐ ........................
☐ ........................

**In addition to the designations made above,** the applicant also makes under Rule 4.9(b) all designations which would be permitted under the PCT except the designation(s) of ________________

The applicant declares that those additional designations are subject to confirmation and that any designation which is not confirmed before the expiration of 15 months from the priority date is to be regarded as withdrawn by the applicant at the expiration of that time limit. *(Confirmation of a designation consists of the filing of a notice specifying that designation and the payment of the designation and confirmation fees. Confirmation must reach the receiving Office within the 15-month time limit.*

Sheet No. ..-4-

**Box No. VI PRIORITY CLAIM** Further priority claims are indicated in the Supplemental Box ☐

The priority of the following earlier application(s) is hereby claimed:

| Country *(in which, or for which, the application was filed)* | Filing Date *(day/month/year)* | Application No. | Office of filing *(only for regional or international application)* |
|---|---|---|---|
| item (1)<br>US | (09.06.1995)<br>09 June 1995 | 08/489,172 | |
| item (2) | | | |
| item (3) | | | |

*Mark the following check-box if the certified copy of the earlier application is to be issued by the Office which for the purposes of the present international application is the receiving Office (a fee may be required):*

☒ The receiving Office is hereby requested to prepare and transmit to the International Bureau a certified copy of the earlier application(s) identified above as item(s) : (1)

**Box No. VII INTERNATIONAL SEARCHING AUTHORITY**

**Choice of International Searching Authority (ISA)** *(if two or more International Searching Authorities are competent to carry out the international search, indicate the Authority chosen; the two-letter code may be used):* ISA / US

**Earlier search** *Fill in where a search (international, international-type or other) by the International Searching Authority has already been carried out or requested and the Authority is now requested to base the international search, to the extent possible, on the results of that earlier search. Identify such search or request either by reference to the relevant application (or the translation thereof) or by reference to the search request:*

Country (or regional Office): Date *(day/month/year)*: Number:

**Box No. VIII CHECK LIST**

This international application contains the following number of sheets:

1. request : 4 sheets
2. description : 30 sheets
3. claims : 12 sheets
4. abstract : 1 sheets
5. drawings : 0 sheets
6. Append: 7

Total : 54 sheets

This international application is accompanied by the item(s) marked below:

1. ☒ separate signed power of attorney
2. ☐ copy of general power of attorney
3. ☐ statement explaining lack of signature
4. ☐ priority document(s) identified in Box No. VI as item(s):
5. ☒ fee calculation sheet
6. ☐ separate indications concerning deposited microorganisms
7. ☐ nucleotide and/or amino acid sequence listing (diskette)
8. ☐ other *(specify)*:

Figure No. ________ of the drawings (if any) should accompany the abstract when it is published.

**Box No. IX SIGNATURE OF APPLICANT OR AGENT**

*Next to each signature, indicate the name of the person signing and the capacity in which the person signs (if such capacity is not obvious from reading the request).*

Frank Pietrantonio

For receiving Office use only

1. Date of actual receipt of the purported international application:
2. Drawings:
   - ☐ received:
   - ☐ not received:
3. Corrected date of actual receipt due to later but timely received papers or drawings completing the purported international application:
4. Date of timely receipt of the required corrections under PCT Article 11(2):
5. International Searching Authority specified by the applicant: ISA /
6. ☐ Transmittal of search copy delayed until search fee is paid

For International Bureau use only

Date of receipt of the record copy by the International Bureau:

**PCT**

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

# INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

| | | |
|---|---|---|
| **(51) International Patent Classification** $^6$ :<br>**H04L 9/00** | **A3** | **(11) International Publication Number:** **WO 96/42151**<br>**(43) International Publication Date:** 27 December 1996 (27.12.96) |

**(21) International Application Number:** PCT/US96/10257

**(22) International Filing Date:** 7 June 1996 (07.06.96)

**(30) Priority Data:**
08/489,172 9 June 1995 (09.06.95) US

**(71) Applicant:** THE DICE COMPANY [US/US]; P.O. Box 60471, Palo Alto, CA 94306-0471 (US).

**(72) Inventors:** COOPERMAN, Marc, S.; 2929 Ramona, Palo Alto, CA 94306 (US). MOSKOWITZ, Scott, A.; Townhouse 4, 20191 East Country Club Drive, North Miami Beach, FL 33180 (US).

**(74) Agents:** ALTMILLER, John, C. et al.; Kenyon & Kenyon, 1025 Connecticut Avenue, N.W., Washington, DC 20036 (US).

**(81) Designated States:** CA, CN, FI, JP, KR, SG, European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

**Published**

*With international search report.*

*Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.*

**(88) Date of publication of the international search report:** 13 February 1997 (13.02.97)

**(54) Title:** STEGANOGRAPHIC METHOD AND DEVICE

**(57) Abstract**

An apparatus and method for encoding and decoding additional information into a stream of digitized samples in an integral manner. The information is encoded using special keys. The information is contained in the samples, not prepended or appended to the sample stream. The method makes it extremely difficult to find the information in the samples if the proper keys are not possessed by the decoder. The method does not cause a significant degradation to the sample stream. The method is used to establish ownership of copyrighted digital multimedia content and provide a disincentive to piracy of such material.

# INTERNATIONAL SEARCH REPORT

International application No.
PCT/US96/10257

## A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) :H04L 9/00
US CL :380/28
According to International Patent Classification (IPC) or to both national classification and IPC

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 380/28; 340/825.34, 4, 23

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

| Category* | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|---|---|---|
| A | US, A, 4,908,873 (PHILIBERT et al) 13 MARCH 1990, See col. 5, lines 1-25. | 1-32 |
| A | US,A, 4,979,210 (NAGATA et al) 18 DECEMBER 1990, See Fig. 13. | 1-32 |
| A | US,A, 5,073,925 (NAGATA et al) 17 DECEMBER 1991, See Fig. 1. | 1-32 |
| A | US,A, 5,287,407 (HOLMES) 15 FEBRUARY 1994, See Fig. 1. | 1-32 |
| A | US,A, 5,365,586 (INDECK et al) 15 NOVEMBER 1994, See cols. 3 and 4. | 1-32 |
| A | US,A, 5,408,505 (INDECK et al) 18 APRIL 1995, See Fig. 4. | 1-32 |

☒ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

| | |
|---|---|
| Date of the actual completion of the international search<br>11 JUNE 1996 | Date of mailing of the international search report<br>24 DEC 1996 |
| Name and mailing address of the ISA/US<br>Commissioner of Patents and Trademarks<br>Box PCT<br>Washington, D.C. 20231<br>Facsimile No. (703) 305-3230 | Authorized officer<br>SALVATORE CANGIALOSI<br>Telephone No. (703) 305-1837 |

Form PCT/ISA/210 (second sheet)(July 1992)*

## INTERNATIONAL SEARCH REPORT

International application No.
PCT/US96/10257

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

| Category* | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|---|---|---|
| A | US,A, 5,412,718 (NARASIMHALU et al) 02 MAY 1995, See Figs. 6A-6C | 1-32 |

Form PCT/ISA/210 (continuation of second sheet)(July 1992)*

# PATENT COOPERATION TREATY

From the
INTERNATIONAL PRELIMINARY EXAMINING AUTHORITY

To: JOHN C. ALTMILLER
KENYON & KENYON
1025 CONNECTICUT AVE. N.W.
WASHINGTON, D.C. 20036

# PCT

## NOTIFICATION OF TRANSMITTAL OF INTERNATIONAL PRELIMINARY EXAMINATION REPORT

(PCT Rule 71.1)

Date of Mailing *(day/month/year)* 25 SEP 1997

Applicant's or agent's file reference
2377/13

**IMPORTANT NOTIFICATION**

| International application No. | International filing date *(day/month/year)* | Priority Date *(day/month/year)* |
|---|---|---|
| PCT/US96/10257 | 07 JUNE 1996 | 07JUNE 1995 |

Applicant
THE DICE COMPANY

1. The applicant is hereby notified that this International Preliminary Examining Authority transmits herewith the international preliminary examination report and its annexes, if any, established on the international application.

2. A copy of the report and its annexes, if any, is being transmitted to the International Bureau for communication to all the elected Offices.

3. Where required by any of the elected Offices, the International Bureau will prepare an English translation of the report (but not of any annexes) and will transmit such translation to those Offices.

4. **REMINDER**

   The applicant must enter the national phase before each elected Office by performing certain acts (filing translations and paying national fees) within 30 months from the priority date (or later in some Offices)(Article 39(1))(see also the reminder sent by the International Bureau with Form PCT/IB/301).

   Where a translation of the international application must be furnished to an elected Office, that translation must contain a translation of any annexes to the international preliminary examination report. It is the applicant's responsibility to prepare and furnish such translation directly to each elected Office concerned.

   For further details on the applicable time limits and requirements of the elected Offices, see Volume II of the PCT Applicant's Guide.

Name and mailing address of the IPEA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231

Facsimile No. (703) 305-3230

Authorized officer
for SALVATORE CANGIALOSI

Telephone No. (703) 305-1837

Form PCT/IPEA/416 (July 1992)*

07-10-07

AG

OIPE IAP87
JUL 0 9 2007
PATENT & TRADEMARK OFFICE

PTO/SB/21 (04-07)
Approved for use through 09/30/2007. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

# TRANSMITTAL FORM

*(to be used for all correspondence after initial filing)*

| | |
|---|---|
| Application Number | 10/049,101 |
| Filing Date | July 23, 2002 |
| First Named Inventor | Scott MOSKOWITZ |
| Art Unit | 2131 |
| Examiner Name | Jeremiah AVERY |
| Attorney Docket Number | 80408.0011 |
| Total Number of Pages in This Submission | |

## ENCLOSURES *(Check all that apply)*

- ☐ Fee Transmittal Form
  - ☐ Fee Attached
- ☑ Amendment/Reply
  - ☑ After Final
  - ☐ Affidavits/declaration(s)
- ☐ Extension of Time Request
- ☐ Express Abandonment Request
- ☐ Information Disclosure Statement
- ☐ Certified Copy of Priority Document(s)
- ☐ Reply to Missing Parts/ Incomplete Application
  - ☐ Reply to Missing Parts under 37 CFR 1.52 or 1.53
- ☐ Drawing(s)
- ☐ Licensing-related Papers
- ☐ Petition
- ☐ Petition to Convert to a Provisional Application
- ☐ Power of Attorney, Revocation Change of Correspondence Address
- ☐ Terminal Disclaimer
- ☐ Request for Refund
- ☐ CD, Number of CD(s) ______
  - ☐ Landscape Table on CD
- ☐ After Allowance Communication to TC
- ☐ Appeal Communication to Board of Appeals and Interferences
- ☐ Appeal Communication to TC **(Appeal Notice, Brief, Reply Brief)**
- ☐ Proprietary Information
- ☐ Status Letter
- ☑ Other Enclosure(s) (please identify below):

Remarks

Copies of EPO Application No. 96919405.9 and Japanese Patent Application No. 2000-542907

## SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

| | | | |
|---|---|---|---|
| Firm Name | | | |
| Signature | Scott Moskowitz | | |
| Printed name | Scott MOSKOWITZ | | |
| Date | July 9, 2007 | Reg. No. | |

## CERTIFICATE OF TRANSMISSION/MAILING

I hereby certify that this correspondence is being facsimile transmitted to the USPTO or deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date shown below:

| | | | |
|---|---|---|---|
| Signature | Scott Moskowitz | | |
| Typed or printed name | Scott MOSKOWITZ | Date | July 9, 2007 |

This collection of information is required by 37 CFR 1.5. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and1.14. This collection is estimated to 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

*If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.*

## PATENT APPLICATION FEE DETERMINATION RECORD
Effective October 1, 2001

Application or Docket Number
10/049101

### CLAIMS AS FILED - PART I

|  | (Column 1) | (Column 2) |
|---|---|---|
| TOTAL CLAIMS |  |  |
| FOR | NUMBER FILED | NUMBER EXTRA |
| TOTAL CHARGEABLE CLAIMS | 30 minus 20= | * ~~+10~~ 11 |
| INDEPENDENT CLAIMS | 7 minus 3 = | * 4 |
| MULTIPLE DEPENDENT CLAIM PRESENT |  | ☐ |

* If the difference in column 1 is less than zero, enter "0" in column 2

SMALL ENTITY TYPE ☐ OR OTHER THAN SMALL ENTITY

| RATE | FEE | | RATE | FEE |
|---|---|---|---|---|
| BASIC FEE | 370 | OR | BASIC FEE |  |
| X$ 9= | 90 | OR | X$18= |  |
| X42= | 168 | OR | X84= |  |
| +140= | / | OR | +280= |  |
| TOTAL | 628 | OR | TOTAL |  |

### CLAIMS AS AMENDED - PART II

7-3-06

| AMENDMENT A |  | (Column 1) CLAIMS REMAINING AFTER AMENDMENT |  | (Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR | (Column 3) PRESENT EXTRA |
|---|---|---|---|---|---|
|  | Total | * 31 | Minus | ** 31 | * 0 |
|  | Independent | * 7 | Minus | *** 7 | * 0 |
|  | FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM |  |  |  | ☐ |

SMALL ENTITY OR OTHER THAN SMALL ENTITY

| RATE | ADDI-TIONAL FEE | | RATE | ADDI-TIONAL FEE |
|---|---|---|---|---|
| X$ 9= | 0 | OR | X$18= |  |
| X42= | 0 | OR | X84= |  |
| +140= | 0 | OR | +280= |  |
| TOTAL ADDIT. FEE | 0 | OR | TOTAL ADDIT. FEE |  |

2-7-07

| AMENDMENT B |  | (Column 1) CLAIMS REMAINING AFTER AMENDMENT |  | (Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR | (Column 3) PRESENT EXTRA |
|---|---|---|---|---|---|
|  | Total | * 31 | Minus | ** 31 | * |
|  | Independent | * 7 | Minus | *** 7 | * |
|  | FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM |  |  |  | ☐ |

| RATE | ADDI-TIONAL FEE | | RATE | ADDI-TIONAL FEE |
|---|---|---|---|---|
| X$ 9= |  | OR | X$18= |  |
| X42= |  | OR | X84= |  |
| +140= |  | OR | +280= |  |
| TOTAL ADDIT. FEE |  | OR | TOTAL ADDIT. FEE |  |

7-9 07

| AMENDMENT C |  | (Column 1) CLAIMS REMAINING AFTER AMENDMENT |  | (Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR | (Column 3) PRESENT EXTRA |
|---|---|---|---|---|---|
|  | Total | * 31 | Minus | ** 31 | * / |
|  | Independent | * 7 | Minus | *** 7 | * / |
|  | FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM |  |  |  | ☐ |

| RATE | ADDI-TIONAL FEE | | RATE | ADDI-TIONAL FEE |
|---|---|---|---|---|
| X$ 9= |  | OR | X$18= |  |
| X42= |  | OR | X84= |  |
| +140= |  | OR | +280= |  |
| TOTAL ADDIT. FEE |  | OR | TOTAL ADDIT. FEE |  |

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20."
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3."
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

FORM PTO-875 (Rev. 8/01)

Patent and Trademark Office, U.S. DEPARTMENT OF COMMERCE

Best Available Copy

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 10/049,101 | 07/23/2002 | Scott A. Moskowitz | 80408.0011 | 8028 |

7590 07/31/2007

Scott A. Moskowitz
#2505
16711 Collins Avenue
Miami, FL 33160

| EXAMINER | |
|---|---|
| AVERY, JEREMIAH L | |

| ART UNIT | PAPER NUMBER |
|---|---|
| 2131 | |

| MAIL DATE | DELIVERY MODE |
|---|---|
| 07/31/2007 | PAPER |

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

PTOL-90A (Rev. 04/07)

80

| | | |
|---|---|---|
| **Advisory Action**<br>**Before the Filing of an Appeal Brief** | **Application No.**<br>*10/049,101* | **Applicant(s)**<br>MOSKOWITZ, SCOTT A. |
| | **Examiner**<br>Jeremiah Avery | **Art Unit**<br>2131 |

***--The MAILING DATE of this communication appears on the cover sheet with the correspondence address --***

THE REPLY FILED 09 July 2007 FAILS TO PLACE THIS APPLICATION IN CONDITION FOR ALLOWANCE.

1. ☐ The reply was filed after a final rejection, but prior to or on the same day as filing a Notice of Appeal. To avoid abandonment of this application, applicant must timely file one of the following replies: (1) an amendment, affidavit, or other evidence, which places the application in condition for allowance; (2) a Notice of Appeal (with appeal fee) in compliance with 37 CFR 41.31; or (3) a Request for Continued Examination (RCE) in compliance with 37 CFR 1.114. The reply must be filed within one of the following time periods:
   a) ☐ The period for reply expires ______months from the mailing date of the final rejection.
   b) ☒ The period for reply expires on: (1) the mailing date of this Advisory Action, or (2) the date set forth in the final rejection, whichever is later. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of the final rejection.
   Examiner Note: If box 1 is checked, check either box (a) or (b). ONLY CHECK BOX (b) WHEN THE FIRST REPLY WAS FILED WITHIN TWO MONTHS OF THE FINAL REJECTION. See MPEP 706.07(f).

Extensions of time may be obtained under 37 CFR 1.136(a). The date on which the petition under 37 CFR 1.136(a) and the appropriate extension fee have been filed is the date for purposes of determining the period of extension and the corresponding amount of the fee. The appropriate extension fee under 37 CFR 1.17(a) is calculated from: (1) the expiration date of the shortened statutory period for reply originally set in the final Office action; or (2) as set forth in (b) above, if checked. Any reply received by the Office later than three months after the mailing date of the final rejection, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

NOTICE OF APPEAL

2. ☐ The Notice of Appeal was filed on _____. A brief in compliance with 37 CFR 41.37 must be filed within two months of the date of filing the Notice of Appeal (37 CFR 41.37(a)), or any extension thereof (37 CFR 41.37(e)), to avoid dismissal of the appeal. Since a Notice of Appeal has been filed, any reply must be filed within the time period set forth in 37 CFR 41.37(a).

AMENDMENTS

3. ☐ The proposed amendment(s) filed after a final rejection, but prior to the date of filing a brief, will not be entered because
   (a) ☐ They raise new issues that would require further consideration and/or search (see NOTE below);
   (b) ☐ They raise the issue of new matter (see NOTE below);
   (c) ☐ They are not deemed to place the application in better form for appeal by materially reducing or simplifying the issues for appeal; and/or
   (d) ☐ They present additional claims without canceling a corresponding number of finally rejected claims.
   NOTE: _____. (See 37 CFR 1.116 and 41.33(a)).
4. ☐ The amendments are not in compliance with 37 CFR 1.121. See attached Notice of Non-Compliant Amendment (PTOL-324).
5. ☐ Applicant's reply has overcome the following rejection(s): _____.
6. ☐ Newly proposed or amended claim(s) _____ would be allowable if submitted in a separate, timely filed amendment canceling the non-allowable claim(s).
7. ☒ For purposes of appeal, the proposed amendment(s): a) ☐ will not be entered, or b) ☒ will be entered and an explanation of how the new or amended claims would be rejected is provided below or appended.
   The status of the claim(s) is (or will be) as follows:
   Claim(s) allowed: _____.
   Claim(s) objected to: _____.
   Claim(s) rejected: *1-31*.
   Claim(s) withdrawn from consideration: _____.

AFFIDAVIT OR OTHER EVIDENCE

8. ☐ The affidavit or other evidence filed after a final action, but before or on the date of filing a Notice of Appeal will not be entered because applicant failed to provide a showing of good and sufficient reasons why the affidavit or other evidence is necessary and was not earlier presented. See 37 CFR 1.116(e).
9. ☐ The affidavit or other evidence filed after the date of filing a Notice of Appeal, but prior to the date of filing a brief, will not be entered because the affidavit or other evidence failed to overcome all rejections under appeal and/or appellant fails to provide a showing a good and sufficient reasons why it is necessary and was not earlier presented. See 37 CFR 41.33(d)(1).
10. ☐ The affidavit or other evidence is entered. An explanation of the status of the claims after entry is below or attached.

REQUEST FOR RECONSIDERATION/OTHER

11. ☒ The request for reconsideration has been considered but does NOT place the application in condition for allowance because: See Continuation Sheet.
12. ☐ Note the attached Information Disclosure Statement(s). (PTO/SB/08) Paper No(s). _____
13. ☐ Other: _____.

CHRISTOPHER REVAK
PRIMARY EXAMINER

Continuation of 11. does NOT place the application in condition for allowance because: Though the Applicant provides further explanation with regards to the terminology found within the claim language (e.g., "legacy content" and "predetermined quality level"), said terminology can possess more than one broad interpretation.
Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See In re Van Geuns, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993). Additional language from the Specification inserted into the claim language and/or supplementary language further elaborating upon said terminology would help to further narrow the level of interpretation of said "legacy content" and "predetermined quality level".

OIPE
JUL 0 9 2007

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

Appl. No. : 10/049,101 Confirmation No. 8028
Applicant : Scott A. Moskowitz, et al.
Filed : July 23, 2002
TC/A.U. : 2131
Examiner : Jeremiah AVERY

Docket No. : 80408.0011

**Mail Stop: After Final**
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

## RESPONSE TO FINAL OFFICE ACTION

In response to the final Office Action of May 9, 2007 Applicants provide the following remarks:

Ok to Enter
JLA 7/25/07

OIPE IAP37
AUG 0 9 2007
PATENT & TRADEMARK OFFICE

8-10-07

RCE/

PTO/SB/30 (04-07)
Approved for use through 09/30/2007. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

# Request for Continued Examination (RCE) Transmittal

Address to:
Mail Stop RCE
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

| | |
|---|---|
| Application Number | 10/049,101 |
| Filing Date | July 23, 2002 |
| First Named Inventor | Scott A. MOSKOWITZ |
| Art Unit | 2131 |
| Examiner Name | Jeremiah L. AVERY |
| Attorney Docket Number | 80408.0011 |

**This is a Request for Continued Examination (RCE) under 37 CFR 1.114 of the above-identified application.**
Request for Continued Examination (RCE) practice under 37 CFR 1.114 does not apply to any utility or plant application filed prior to June 8, 1995, or to any design application. See Instruction Sheet for RCEs (not to be submitted to the USPTO) on page 2.

1. Submission required under 37 CFR 1.114 Note: If the RCE is proper, any previously filed unentered amendments and amendments enclosed with the RCE will be entered in the order in which they were filed unless applicant instructs otherwise. If applicant does not wish to have any previously filed unentered amendment(s) entered, applicant must request non-entry of such amendment(s).
   - a. ☐ Previously submitted. If a final Office action is outstanding, any amendments filed after the final Office action may be considered as a submission even if this box is not checked.
     - i. ☐ Consider the arguments in the Appeal Brief or Reply Brief previously filed on ____________
     - ii. ☐ Other ____________
   - b. ☑ Enclosed
     - i. ☑ Amendment/Reply
     - ii. ☐ Affidavit(s)/ Declaration(s)
     - iii. ☐ Information Disclosure Statement (IDS)
     - iv. ☐ Other ____________
2. Miscellaneous
   - a. ☐ Suspension of action on the above-identified application is requested under 37 CFR 1.103(c) for a period of _______ months. (Period of suspension shall not exceed 3 months; Fee under 37 CFR 1.17(i) required)
   - b. ☐ Other ____________
3. Fees The RCE fee under 37 CFR 1.17(e) is required by 37 CFR 1.114 when the RCE is filed.
   - a. ☐ The Director is hereby authorized to charge the following fees, any underpayment of fees, or credit any overpayments, to Deposit Account No. ____________. I have enclosed a duplicate copy of this sheet.
     - i. ☐ RCE fee required under 37 CFR 1.17(e)
     - ii. ☐ Extension of time fee (37 CFR 1.136 and 1.17)
     - iii. ☐ Other ____________
   - b. ☐ Check in the amount of $ ____________ enclosed
   - c. ☑ Payment by credit card (Form PTO-2038 enclosed)

**WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.**

### SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT REQUIRED

| | | | |
|---|---|---|---|
| Signature | | Date | August 9, 2007 |
| Name (Print/Type) | Scott A. MOSKOWITZ | Registration No. | |

### CERTIFICATE OF MAILING OR TRANSMISSION

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Mail Stop RCE, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450 or facsimile transmitted to the U.S. Patent and Trademark Office on the date shown below.

| | | | |
|---|---|---|---|
| Signature | | | |
| Name (Print/Type) | Scott A. MOSKOWITZ | Date | August 9, 2007 |

This collection of information is required by 37 CFR 1.114. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Mail Stop RCE, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

*If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.*

08/13/2007 HLE333 00000005 10049101
01 FC:2801 395.00 OP

OIPE
AUG 0 9 2007
PATENT & TRADEMARK OFFICE

PTO/SB/17 (07-07)
Approved for use through 06/30/2010. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number

*Effective on 12/08/2004.*
*Fees pursuant to the Consolidated Appropriations Act, 2005 (H.R. 4818).*

# FEE TRANSMITTAL
## For FY 2007

☑ Applicant claims small entity status. See 37 CFR 1.27

**TOTAL AMOUNT OF PAYMENT** ($) $395.00

***Complete if Known***

| | |
|---|---|
| Application Number | 10/049,101 |
| Filing Date | July 23, 2002 |
| First Named Inventor | Scott A. MOSKOWITZ |
| Examiner Name | Jeremiah L. AVERY |
| Art Unit | 2131 |
| Attorney Docket No. | 80408.0011 |

**METHOD OF PAYMENT** (check all that apply)

☐ Check ☑ Credit Card ☐ Money Order ☐ None ☐ Other (please identify): ____________

☐ Deposit Account Deposit Account Number: ____________ Deposit Account Name: ____________

For the above-identified deposit account, the Director is hereby authorized to: (check all that apply)

☐ Charge fee(s) indicated below ☐ Charge fee(s) indicated below, **except for the filing fee**

☐ Charge any additional fee(s) or underpayments of fee(s) under 37 CFR 1.16 and 1.17 ☐ Credit any overpayments

**WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.**

**FEE CALCULATION**

**1. BASIC FILING, SEARCH, AND EXAMINATION FEES**

| | FILING FEES | | SEARCH FEES | | EXAMINATION FEES | | |
|---|---|---|---|---|---|---|---|
| **Application Type** | **Fee ($)** | **Small Entity Fee ($)** | **Fee ($)** | **Small Entity Fee ($)** | **Fee ($)** | **Small Entity Fee ($)** | **Fees Paid ($)** |
| Utility | 300 | 150 | 500 | 250 | 200 | 100 | |
| Design | 200 | 100 | 100 | 50 | 130 | 65 | |
| Plant | 200 | 100 | 300 | 150 | 160 | 80 | |
| Reissue | 300 | 150 | 500 | 250 | 600 | 300 | |
| Provisional | 200 | 100 | 0 | 0 | 0 | 0 | |

**2. EXCESS CLAIM FEES**

| **Fee Description** | **Fee ($)** | **Small Entity Fee ($)** |
|---|---|---|
| Each claim over 20 (including Reissues) | 50 | 25 |
| Each independent claim over 3 (including Reissues) | 200 | 100 |
| Multiple dependent claims | 360 | 180 |

**Total Claims** **Extra Claims** **Fee ($)** **Fee Paid ($)**
________ - 20 or HP = ________ x ________ = ________
HP = highest number of total claims paid for, if greater than 20.

**Indep. Claims** **Extra Claims** **Fee ($)** **Fee Paid ($)**
________ - 3 or HP = ________ x ________ = ________
HP = highest number of independent claims paid for, if greater than 3.

**Multiple Dependent Claims**
**Fee ($)** **Fee Paid ($)**
________ ________

**3. APPLICATION SIZE FEE**

If the specification and drawings exceed 100 sheets of paper (excluding electronically filed sequence or computer listings under 37 CFR 1.52(e)), the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).

**Total Sheets** **Extra Sheets** **Number of each additional 50 or fraction thereof** **Fee ($)** **Fee Paid ($)**
________ - 100 = ________ / 50 = ________ (round up to a whole number) x ________ = ________

**4. OTHER FEE(S)** **Fees Paid ($)**

Non-English Specification, $130 fee (no small entity discount) ________

Other (e.g., late filing surcharge): REQUEST FOR CONTINUED EXAMINATION ("RCE") $395.00

**SUBMITTED BY**

| | | | |
|---|---|---|---|
| Signature | | Registration No. (Attorney/Agent) | Telephone 305 956 9041 |
| Name (Print/Type) | Scott A. MOSKOWITZ | | Date August 9, 2007 |

This collection of information is required by 37 CFR 1.136. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 30 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

*If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.*

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In RE: Application of:

Scott A. Moskowitz, et al. ) Examiner: Jeremiah L. AVERY

Application No.: 10/049,101 ) Group Art: 2131

Filed: July 23, 2002

For: A SECURE PERSONAL CONTENT SERVER

**Mail Stop: After Final / Request for Continued Examination ("RCE")**
Assistant Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

# REQUEST FOR CONTINUED EXAMINATION ("RCE") UNDER 37 C.F.R. § 1.114

In response to the final Office Action of May 9, 2007 and the July 31, 2007 Advisory Action Applicants respectfully submit herewith a Request for Continued Examination ("RCE"). The Applicants respectfully request the Office to reconsider the application in view of the following remarks:

**In the Claims:**

Applicants reserve the right to pursue the subject matter of the original claims in this application and in other applications. This listing of claims will replace all prior versions, and listings, of claims in the application:

**Listing of Claims:**

1. (previously presented) A local content server system (LCS) for creating a secure environment for digital content, comprising:

a) a communications port in communication for connecting the system via a network to at least one Secure Electronic Content Distributor (SECD), said SECD capable of storing a plurality of data sets, capable of receiving a request to transfer at least one content data set, and capable of transmitting the at least one content data set in a secured transmission;

b) a rewritable storage medium whereby content received from outside the LCS may be stored and retrieved;

c) a domain processor that imposes rules and procedures for content being transferred between the LCS and devices outside the LCS; and

d) a programmable address module which can be programmed with an identification code uniquely associated with the LCS; and

said domain processor permitting the LCS to receive digital content from outside the LCS provided the LCS first determines that the digital content being delivered to the LCS is authorized for use by the LCS and if the digital content is not authorized for use by the LCS, accepting the digital content at a predetermined quality level, said predetermined quality level having been set for legacy content.

2. (original) The LCS of claim 1 further comprising

e) an interface to permit the LCS to communicate with one or more Satellite Units (SU) which may be connected to the system through the interface, said SUs capable of receiving and transmitting digital content;

and wherein said domain processor permits the LCS to receive digital content from an SECD that is connected to the LCS's communication port, provided the LCS first determines that digital content being received is authorized for use by the LCS,

and wherein said domain processor permits the LCS to deliver digital content to an SU that may be connected to the LCS's interface, provided the LCS first determines that digital content being received is authorized for use by the SU.

3. (previously presented) A local content server system (LCS) for creating a secure environment for digital content, comprising:

a) a communications port in communication for connecting the system via a network to at least one Secure Electronic Content Distributor (SECD), said SECD capable of storing a plurality of data sets, capable of receiving a request to transfer at least one content data set, and capable of transmitting the at least one content data set in a secured transmission;

b) an interface to permit the LCS to communicate with one or more Satellite Units (SU) which may be connected to the system through the interface, said SUs capable of receiving and transmitting digital content; and

c) a rewritable storage medium whereby content received from an SECD and from an SU may be stored and retrieved;

d) a domain processor that imposes rules and procedures for content being transferred between the LCS and the SECD and between the LCS and the SU; and

e) a programmable address module which can be programmed with an identification code uniquely associated with the LCS;

said domain processor permitting the LCS to deliver digital content to and receive digital content from an SU that is connected to the LCS's interface, provided the LCS first determines that the digital content being delivered to the SU is authorized for use by the SU or that the digital content being received is authorized for use by the LCS, and if the digital content is not authorized for use, accepting the digital content at a predetermined quality level, said predetermined quality level having been set for legacy content,

and said domain processor permitting the LCS to receive digital content from an SECD that is connected to the LCS's communication port, provided the LCS first determines that digital content being received is authorized for use by the LCS and if the digital content is not authorized for use by the LCS, accepting the digital content at a predetermined quality level, said predetermined quality level having been set for legacy content.

4. (original) The system of claim 3, wherein said domain processor determines whether digital content is authorized for use by extracting a watermark from the digital content being transferred.

5. (original) The system of claim 3, wherein said domain processor comprises:

means for obtaining an identification code from an SU connected to the LCS's interface;

an analyzer to analyze the identification code from the SU to determine if the SU is an authorized device for communicating with the LCS;

means for analyzing digital content received from an SU;

said system permitting the digital content to be stored in the LCS if i) an analysis of the digital content received from the SU concludes that the content is authenticated, or ii) an analysis of the digital content received from the SU concludes that the content cannot be authenticated because no authentication data is embedded in the content, and

said system preventing the digital content from being stored on the LCS if i) an analysis of the digital content received from the SU concludes that the content is unauthenticated.

6. (original) The system of claim 4, wherein said analyzer of the domain processor comprises means for extracting digital watermarks from the digital content received from an SU, and means for analyzing the digital watermark to determine if the digital content has been previously marked with the unique identification code of the LCS.

7. (original) The system of claim 4, wherein said system permits the digital content to be stored in the LCS at a degraded quality level if an analysis of the digital content received from the SU concludes that the digital content received from the SU cannot be authenticated because there is no authentication data embedded in the content.

8. (original) The system of claim 4, further comprising at least one SU, each such SU being capable of communicating with the LCS.

9. (original) The system of claim 8, wherein the SU has means to sending a message to the LCS indicating that the SU is requesting a copy of a content data set that is stored on the LCS, said message including information about the identity of the SU, and wherein the LCS comprises:

means to analyze the message from the SU to confirm that the SU is authorized to use the LCS;

means to retrieve a copy of the requested content data set;

means to embed at least one robust open watermark into the copy of the requested content data set, said watermark indicating that the copy is authenticated;

means to embed a second watermark into the copy of the requested content data set, said second watermark being created based upon information transmitted by the SU and information about the LCS; and

means to deliver the watermarked content data set to the SU for its use.