Multimedia Content Protection in the 21st Century

Rich Goyette

10 Dec 04

This document was prepared for Professor A. Karmouch in partial fulfillment of the requirements for the course ELG5199 Design of Multimedia Distributed Database Systems

Abstract

In this paper, we review techniques and processes for securing multimedia content. Specifically, we examine encryption and watermarking techniques as these are currently seen as the most effective (yet complementary) approaches to content security.

We consider three approaches for providing confidentiality services (through encryption) in conditional access system applications and examine the difficulties and issues arising out of combining encryption with varying bandwidth and end-user device capabilities. Since key management is generally not covered in literature dealing with encryption, we examine some interesting key management and distribution architectures. A basic key management approach is examined, followed by a system based on key graphs. We finish with an examination of a system using shared secrets instead of symmetric or asymmetric keys.

Watermarking definitions and properties are reviewed and the field of watermarking is examined in a general way. We research some attacks on watermarks since these tend to drive research efforts.

We conclude this paper with a high level examination of Digital Rights Management (DRM). This examination includes elements that comprise a DRM system, some of the grass-roots (OMA MDRM) and top down (MPEG-21) approaches towards implementing open and interoperable DRM, and some legal approaches that have been enacted to address digital rights.

Table of Contents

1 Introduction

The migration of media from an analog to a digital world has presented the creators of these works with an unfortunate double-edged sword. On one hand, it has opened up many new and potentially lucrative multimedia digital service markets such as Video On Demand (VOD), Pay-per-View, and real time gaming [7],[4],[1] by providing an efficient and timely means of content delivery over public switched networks.

On the other hand, "the easy replication of digital media, otherwise an advantage, is a major concern in the context of copyright infringements." ([15], p. 444). We have seen manifestations of this recently as the Recording Industry Association of America (RIAA) attempts to subdue users of popular Peer-to-Peer file sharing networks (Kazaa, Napster, etc) through civil legal attacks. Although these file-sharing networks are associated primarily with music, they are also used to share other multimedia content as well – including movies and video games.

It is therefore generally acknowledged that "because of its high economic value, copyrighted entertainment content needs to be protected [and that] end-to-end security is the most critical requirement for the creation of new digital markets." ([3], p. 238). This relatively new requirement has been tacitly acknowledged in the literature; there have been a number of special issues on security (for example, MultiMedia Security 2003) and the annual ACM multimedia conference was renamed to Multimedia and *Security*. Perusing the literature, one finds an increasing number of works directed at solving the problems of securing digital media on open networks.

Since it is one of the major motivations behind the desire to secure content, we begin this paper with a brief review of the forces that drive underground Internet Piracy. We then focus on the difficult problem of securing multimedia *content in transit* with an analysis of the some of the overhead costs, open issues, and research directions. This is followed by a similar discussion of *content copyright protection* using digital watermarking. We conclude with a generic discussion of *content copy-control* through Digital Rights Management (DRM) schemes and associated standards.

2 Internet Digital Piracy – A Perspective

In this section, we briefly examine the mechanisms and motivations behind digital piracy on the internet with the aim of shedding some light on why there is currently a need for digital content protection and rights management.

Over the relatively short lifetime of the internet, an entire culture has developed around the illicit trading of copyrighted materials. It began in earnest in the early days computer gaming (during the age of such computers as the Apple and the Commodor Vic 20). Game programs could fit on one or more floppy disks and, because it was easy to copy a disk and duplicate a game, games were shared among friends or small communities. The advent and wide-spread usage of dial-in bulletin boards added another dimension to the sharing of games – a game file could be "uploaded" to a bulletin board in one geographic location and shared quite quickly with others very far away. However, sharing of files required someone with knowledge of the "right" bulletin boards to visit

Soon, primitive copy-protection schemes were developed to protect the software on gaming disks. This touched off an "arms race" of sorts between "crackers" (more commonly known to the media as "hackers") and game distributors. Protection mechanisms were developed, were good for a (very) short period of time, were defeated by crackers, and were replaced by new mechanisms. Information about new "cracks" was also disseminated widely across bulletin boards.

The decline of bulletin boards and the rise of the internet as a means of disseminating information also spawned a new community known as "warez groups" [47] (pronounced the same as "wares" as in "softwares). This community existed (and still exists) solely to freely propagate digital content. People participating in the community refer to themselves as "warez d00dz" (or wares dudes).

As time passed and bandwidth increased, all manner of copyrighted or otherwise protected material made it's way into the "warez scene" – including such content as digitally recorded music and feature length movies. In fact, it is not uncommon to hear about feature length movies, music albums, and video games being released on the internet days or weeks in advance of their public release dates.

In most cases, those making the pirated material available on the internet are not compensated in a traditional monetary fashion. Rather, they compete for the status of being "elite"; their modus operandi has more to do with ego and status than direct monetary remuneration. As described in The New Hacker's Dictionary, "The studly thing to do if one is a warez d00d, it appears, is emit *0-day warez*, that is copies of commercial software copied and cracked on the same day as its retail release." ([47] p. 478).

In the Summer 2004 edition of *2600: The Hacker Quarterly* [48], a thoroughly fascinating article entitled "A Guide to Internet Piracy" provides a glimpse into the world

of digital piracy. The article identifies "Warez/Release Groups" and "Site Traders" as the top two entities in the piracy "food chain." In almost all cases, materials are made available in the public domain by Release Groups. Naturally, release groups are paranoid. They know that the authorities are watching and they go to great lengths to protect themselves. However, these groups need to "brag" about their exploits so each release also includes "*.nfo" (info or information) files that advertise or "brag" the Group [49].

Site Traders, on the other hand, deal only with content that has already been released. Site Traders are identified in [48] as individuals who "race" new releases between warez sites. They do this to gain download "credits" (and associated bragging rights) at each site.

The direct benefactors of the "chest-thumping" efforts of Release Groups, Site Traders, and other entities in the piracy food chain are otherwise normal internet users who partake in the "Darknet." The Darknet is defined in [18] as the "distribution network that emerges from the injection of objects … and the distribution of those objects" ([18] p. 2). The warez community and users of Peer-to-Peer (P2P) networks (such as those popularized by Napster and Kazaa) both inject and consume objects on the darknet at rates undreamed of twenty years ago and with an ease that is almost automatic. Most internet users of P2P software are either unaware or uninterested in the fact that their actions could be violating copyright laws. Combined with the widespread use of portable content copying and rendering devices (DVD burners, MP3 players), there is little incentive for users of the Darknet to actually pay for the material they consume.

A central premise in [18] is that there will always be a small fraction of internet users who will be interested in obtaining digital content in an unprotected, reproducible form (either through the breaking of content protection mechanisms or through access to the material before content protections are applied). As we saw above, there are indeed persons and groups who exist in seemingly well-organized hierarchies that are interested in this type of activity. And, because they make their material available to the Darknet for general consumption (for free), it is very difficult to envision a market strategy that can compete successfully (on a price-comparison basis) to put them out of business.

It is logical, therefore, to consider technologies and techniques that show promise for protecting multimedia content against unremunerated distribution. In this paper, we consider, in a very general fashion, some of these approaches.

3 Multimedia Content Protection in Transit

3.1 Introduction

In this section, we begin looking at multimedia content protection. Specifically, we focus on content protection while it is in transit and we further consider streaming video as the content distribution mechanism. Streaming video is typically the multimedia content of choice for Conditional Access (CA) systems. A CA system, in its most generic definition, is any system that restricts access to multimedia content to a group of (possibly time-varying) end-entities that are authorized to view or access the content. In typical CA systems such as Video-on-Demand and Pay-per-View, authorized end-entities are viewers who have paid a subscription or viewing fee. CA systems have come to dominate the broadcast domain and are currently the only widely deployed form of content protection employed in that domain [28].

We define the security problem in this section to be one of providing confidentiality services to streaming video through **digital encryption**. It has been asserted that the only currently useful technical means of providing confidentiality services to streaming video (or any media) is through encryption [3][22]. The use of encryption for confidentiality of streaming video exposes two problem domains that are currently active research areas. The first problem is that the encryption process must not break as a result of (or interfere with) data transformation, data rate scaling, or data transport. Eskicioglu et al. assert that "since the [video] content will often be compressed using a scalable compression technique and transported over a lossy packet network using the Internet Protocol (IP), … security measures must not only be compatible with the compression technique and data transport but be robust to errors as well." ([3] p. 254)

The second issue involves key management and distribution. The security of the system is inherently related to the ways in which the keys are generated, distributed, stored, and used. As we shall see, a CA system with a large and time-varying user base faces non-trivial key management issues. Research efforts must treat both the encryption problem and the key management problem as a single problem [19].

In the remainder of this section, we examine some techniques for the protection of streaming video through encryption and the implication these techniques have for scalability and overhead. We then examine some interesting techniques for key management in conditional access systems. We then present some concluding remarks. However, in order to fully understand some of the approaches taken, we begin with an overview of definitions and terminology.

3.2 Definitions and Terminology

3.2.1 Digital Content

In this section (and throughout the rest of this paper), we define digital content to be any multimedia work that can be copyrighted. Multimedia can include text, audio, video, imagery or any combination of these.

3.2.2 Video Compression or *Coding*

Video compression (or coding) is the process of transforming multimedia (typically still frame images, music, or video) in such a way as to reduce the bandwidth or storage requirement necessary to transmit or retain the media. For example, consider standard NTSC television encoding. The data rate can easily exceed 20 MBytes/s (480 lines with 720 pixels per line and a frame rate of 30 frames per second) [24][5]. Obviously, this is most certainly not an acceptable data rate for most internet connected devices.

Coding can effectively reduce the data rate or storage requirements for image or video data. Coding can be (and usually is) "lossy" in the sense that information is irrevocably lost during the coding process. Depending on the coding standard and mechanisms involved, a tradeoff can be made between compression ratio and image or video quality.

3.2.3 Video Transcoding

Video coders that produce a video stream at a constant bit rate can be problematic in that "the Internet, in particular, is full of uncertainties with a wide range of channel capacities and client device capabilities for display, computation, and communication." ([5] p. 1) Without any other means of mediation, a video stream produced by a video coder for a multicast application would need to be encoded to the lowest channel or processing capacity likely to be encountered anywhere between itself and any client device. It would be useful to encode the video stream to the maximum possible quality and bit rate and then be able to modify the video stream in-transit to meet varying bandwidth requirements or client device processing capabilities.

Video **transcoding** seeks to provide this functionality. In its simplest terms (summarized here from [16]), a transcoder receives an incoming video stream and may perform one or both of the following operations:

- Bit-Rate Reduction: the transcoder attempts to match the bitrate of the incoming stream to the available outbound bandwidth without reducing the quality of the original stream; and

- Spatial and/or Temporal Resolution Reduction: the transcoder performs spatial and/or temporal resolution reduction on the stream content in order to match the video processing capabilities of a class of target devices (for example, PDAs, cell phones). Spatial resolution reduction can be achieved by decoding the input stream, performing a down-sampling of each frame, and then recoding the stream (at a possibly different out-bound rate). Temporal resolution reduction can be achieved by reducing the frame rate of the incoming stream.

3.2.4 Scalable Coding

Scalable coding aims at removing the need for intermediate transcoding of the video stream. In fact, "the holy grail of scalable video coding is to encode the video once and then by simply truncating certain layers or bits from the original stream, lower qualities, spatial resolutions, and/or temporal resolutions could be obtained." ([16] p. 26). Network transport or end user devices would transmit or use as much of the scaled video stream as the channel or device could handle.

Scalable coding and transcoding techniques are simply different approaches to the same problem: "Scalable coding specifies the data format at the encoding stage independently of the transmission requirements, while transcoding converts the existing data format to meet the current transmission requirements." ([16] p. 26).

Both MPEG-2 and MPEG-4 (discussed shortly) include scalability profiles. Scalability is achieved in both standards by producing a "base layer" and one or more "enhancement layers". The base layer contains frame data that has been scaled to some minimum bandwidth and quality which is guaranteed for all network connections and device capabilities. The enhancement layer(s) are composed of the residual information and is/are used to increase the spatial and/or temporal quality of the video stream. The enhancement layers are transmitted on a best effort basis, giving priority to any available bandwidth to the base layer (which, if scoped properly for the application, will have little trouble getting through).

3.2.5 Motion Picture Expert Group (MPEG) Standards

The Motion Picture Experts Group, or MPEG, is an International Standard Organization (ISO) working group whose mandate is to develop "international standards for compression, decompression, processing and coded representation of moving pictures, audio and their combination in order to satisfy a wide variety of applications." ([50] p. 1). It has been developing standards for digital audio and video formatting and compression since its first meeting in Ottawa, Canada in 1998.

Because MPEG standards are so ubiquitous in the storage and transmission of multimedia content – particularly video and audio, we give a short overview of these standards below. Note that MPEG-7 and MPEG-21 are not included in this list as the goal for both

standards has evolved away from digital compression. We cover MPEG-21 later. The material in this abbreviated summary was drawn from [21][24] and [50].

- **MPEG-1**: This was the first standard developed by MPEG and the major parts were approved in November 1992. It was developed for the coding of non-interlaced (or progressive) video up to 1.5 Mbit/sec and audio up to 384 kbit/s. MPEG-1 is extremely pervasive in today's world. MPEG-1 Layer III audio – better known as MP3 – "has given rise to innovative ways of consuming music… With the arrival of MP3 the music world has been changed without recognition." ([50] p. 2).

- **MPEG-2**: This standard, whose first major parts were approved in 1994, is an extension of MPEG-1. It targets interlaced video used in broadcasting formats at compressed rates of up to 15 Mbit/sec and High Definition TV (HDTV) at rates of up to 30 Mbit/set [24]. Starting in MPEG-2, the concept of "profiles" and "levels" are introduced in order to address implementation issues. Specifically, it is generally inefficient for vendors to attempt to build coders and decoders that comply with the full standard and offer all of the associated standards tools. Instead, vendors can declare that their product is compliant with a certain profile and level and this automatically defines the product's capabilities with respect to the MPEG standard [24].

- **MPEG-4**: Version 2 of this standard was approved in 1999 and targets video data rates of up to 2 Gbits/s [51]. As indicated in [50], MPEG-4 is very similar to MPEG-2 for the first six parts of the standard. However, an important enhancement to the standard is that it allows for the representation of individual objects within frames.

3.2.6 Joint Picture Experts Group (JPEG) Standards

JPEG, like MPEG, is also an ISO working group. It was formed starting in 1982 to target the compression of still images – both black and white and colour. It shares some of the same techniques as MPEG for image compression – it has functional blocks for DCT, quantization, and coding ([46] slide 24).

A critique of the baseline JPEG standard [25] points out that the quality of an image obtained at decompression time is determined at compression time (that is, there is no freedom for the decoder to, say, decompress to a lower quality image). JPEG2000, an enhanced ISO JPEG standard, resolves this problem by delaying "the decision on several key compression parameters such as quality or resolution… until after the creation of the codestream" ([25] p. 268). JPEG2000 offers a number of progressive or scalable transmission modes which include (from [25]) *quality* (as more data is received in a transmission, the quality of a baseline image increases) and *resolution* (as more data is received in a transmission, the image size increases) modes.

3.2.7 MPEG Video Compression Mini-Tutorial

Here we provide a brief overview of how image and video compression is achieved using the MPEG standards (those elements that are common to MPEG-1, 2 and 4). Much of the work that follows discusses research involving details of the compression process. It is therefore useful to have a good working knowledge of compression terminology and mechanisms before reviewing that research. Figure 1 shows a simplified block diagram of the basic steps in producing a compressed bit stream. The figure and accompanying description are based on material in [45][46], and[24].

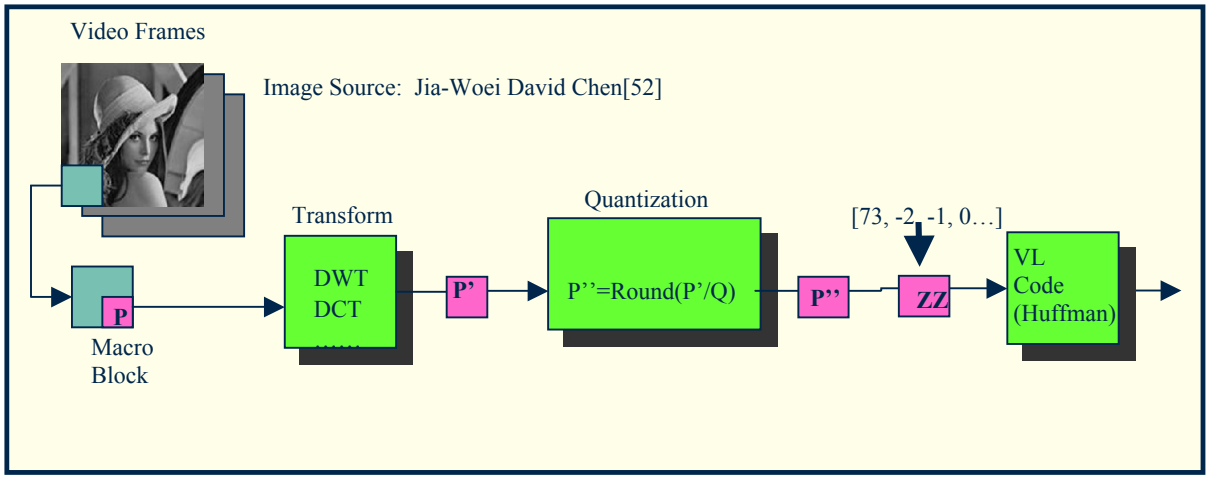

Figure 1: Intra-Frame MPEG Compression

i.) Frame Division: In this phase, an incoming stream of video frames is divided into groupings that are referred to throughout the MPEG standards as MacroBlocks, or MBs. MacroBlocks are typically 16X16 blocks of bits.

ii.) Transformation: In this block, the image data is either *spatially* compressed using a transform (such as a wavelet [2] or cosine [24] transform) or *temporally* compressed using a combination of transform and motion vectors. Spatial compression is referred to as *intra-frame* compression while temporal compression is known as *inter-frame* compression. These two modes are briefly discussed below.

- *Intra*-Frame compression is performed using the Discrete Cosine Transform (DCT) on 8X8 sections of the data. For image and video data, a DCT generally produces a sparse matrix of signed AC and DC coefficients. Figure 2 below shows the result of performing a DCT on the 8X8 block of grayscale data **P** from the Lena image of Figure 1 above. The resulting DCT matrix is referred to as **P'**. Notice how most of the information in the matrix is now concentrated in or about the DC component (the element **P'[0,0]**).

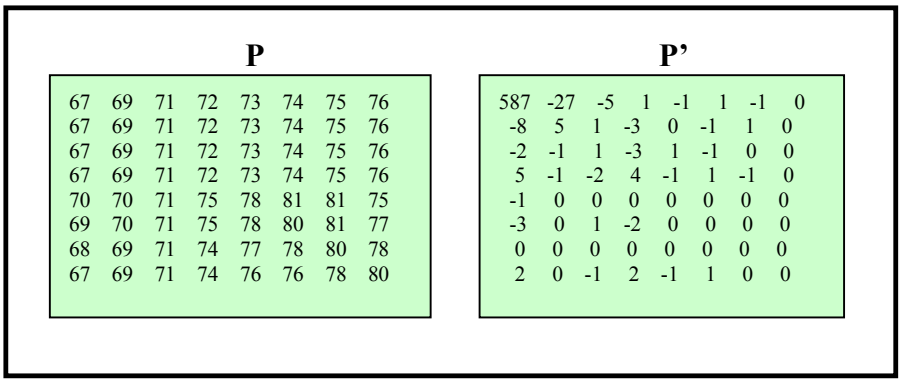

Figure 2: DCT performed on 8X8 Segment of Image Data

- *Inter*-Frame compression is performed through motion compensated predictive compression ([45, slide 47). In many cases throughout a video sequence, very little of a frame actually changes between the last frame and the next frame (for example, a scene that is slowly panning or a newscaster that is speaking) and it is this temporal redundancy that inter-frame predictive compression seeks to capitalize on. While a detailed analysis of motion predictive compression is beyond the scope of this work, it is instructive to elaborate on the essential elements of this mechanism as some of the tactics for securing multimedia (video) streams target these mechanisms. The MPEG standards define three frame types [24][45]:

    - I-Frames: these frames provide a reference for the video stream. Each MB in these frames is intra-frame compressed (that is, spatially compressed using DCT).

    - P-Frames: the MBs in these frames are either motion compensated forward predicted from previous I or P frames or they are Intra-Frame compressed. A general description (from [45]) of a P-Frame generation follows:

        - Refer to Figure 3 below. The MPEG video coder compares each MB from the frame being coded at time (t) to the MBs in a reference frame at (t-1) using one of a number of motion estimation algorithms [46] (for example, best fit block matching).

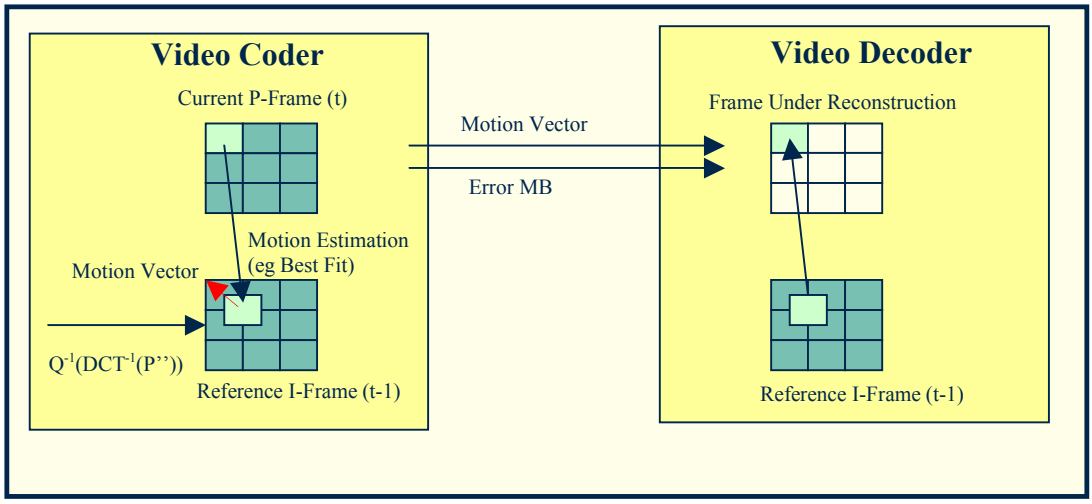

Figure 3: P-Frame Generation

- If a suitable MB in the reference frame is found, a motion vector that maps the MB from the reference frame to the frame being coded is generated. The reference MB may not be a perfect match so an error MB is generated as well. The motion vector and error MB are sent to the decoder. The decoder reproduces each MB in frame (t) by taking MBs from the reference frame and applying the motion vector and error MBs.

- In some cases, no suitable MB will be found in the reference frame (the error MB will be greater than some threshold). In that case, the coder simply intra-codes the MB in frame (t) and sends this instead. If the number of MB being intra-coded is large, this probably indicates that the video sequence has changed significantly (the scene switches completely) and a new reference frame is sent.

- In the figure, the reference frame used by the coder is represented as being inverse quantized and inverse transformed. The coder must use the same reference frame as seen by the decoder and, since the quantization process is lossy, the reference frame must be quantized and inverse quantized in order to be in possession of the same reference frame.

- B-Frames: In this case, the MBs in the frames being reconstructed are predicted using the previous and/or next frame in the video sequence and result in the highest compression rate [24]. However, they do require that the frames be sent by the coder out of sequence since the decoder must have all of the frames necessary to rebuild the B-Frame.

iii.) Quantization: As stated by Tudor, "it has been observed that the numerical precision of the DCT coefficients may be reduced while still maintaining good image quality at the decoder" ([24] p. 259). Quantization is performed on the DCT coefficients to reduce the number of bits necessary to represent them. Quantization is a lossy operation since the quantization error is not recoverable at the decoder. However, as will be discussed later, scalable compression methods have been developed to send the quantization error to the decoder as a "take-it-or-leave-it" enhancement to the already transmitted frame. Quantization is not normally uniformly applied to an 8X8 block (although it is possible). For example, the default quantization matrix for MPEG-1 is shown in Figure 4 ([45, slide 35). Also shown is the result of quantizing **P'** from Figure 3. The quantization matrix for MPEG-4 can be adaptively configured by the coder.

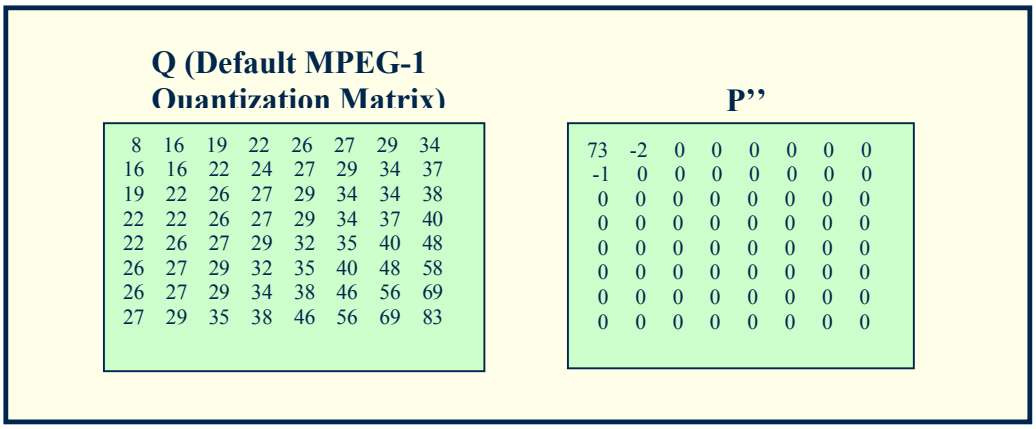

Figure 4: Default quantization matrix for MPEG-1 and quantization of P'.

iv.) Coding: As indicated earlier, a DC transform on an image segment generally results in a sparse matrix and coefficients are often further removed by quantization. The remaining coefficients are then scanned into a linear vector from the 8X8 quantized DCT coefficient matrix in a zig-zag fashion starting at the DC coefficient (P''[0,0]). The scan pattern attempts to place the non-zero coefficients first since they tend to be distributed around the DC component. Each vector is then coded using Huffman-like coding which results in a smaller coded stream. For the running example of Figure 1, the resulting linear vector is [73, –2, –1, 0, 0, 0, ….].

3.3 Review of Stream Encryption Techniques

Now that definitions and terminology are understood, we next examine three approaches to encrypting video streams in order to provide confidentiality and conditional access services. In all three approaches, standard keyed encryption schemes are employed. However, the difficult problem of key management and distribution has been "assumed out" of these solutions. We look at approaches to key management later.

3.3.1 Secure Scalable Video Streaming

This approach is described in [11] and [12]. The authors propose a novel system for adding a measure of scalability and security to transcoding operations. They call their approach Secure Scalable Streaming (SSS). In providing scalability to transcoding, some of the challenging problems of securing a transcoded stream can be solved.

The authors assert that current transcoding algorithms may not be up to the task of processing the hundreds or thousands of high quality video streams that might be found on a major transcoding network node. They also point out that transcoding the stream between a server and a client requires the transcoder to be in possession of any and all secret keys used to secure the streaming content and this introduces a security point of failure.

The scheme is straightforward. It uses scalable coding tools already present in compression standards to build scalable video streams – that is, the data at the start of the stream is used to assemble a "baseline" image or video frame which is then enhanced (in terms of resolution or quality) by data found later in the stream (MPEG-4 FGS SNR [10] for example). The stream is then progressively encrypted so that the entire stream is not needed at the client prior to decryption. Finally, the stream is appended with a clear-text header indicating a number of offset points where the stream could be truncated by a transcoder should the stream need to be adjusted to a different quality or bit rate.

The authors use Motion-JPEG2000 for their scalable coding standard. Motion JPEG is a JPEG2000 based implementation for video streaming. The implementation is shown pictorially in Figure 5. Video frame data is broken down into tiles and coded into scalable streams using JPEG2000. The stream for each tile is then fed to an SSS packer whose task is to:

a.) Extract coding data from the JPEG2000 stream header;
b.) Progressively encrypt the JPEG2000 compressed data; and
c.) Build SSS packets with clear-text headers and encrypted data.

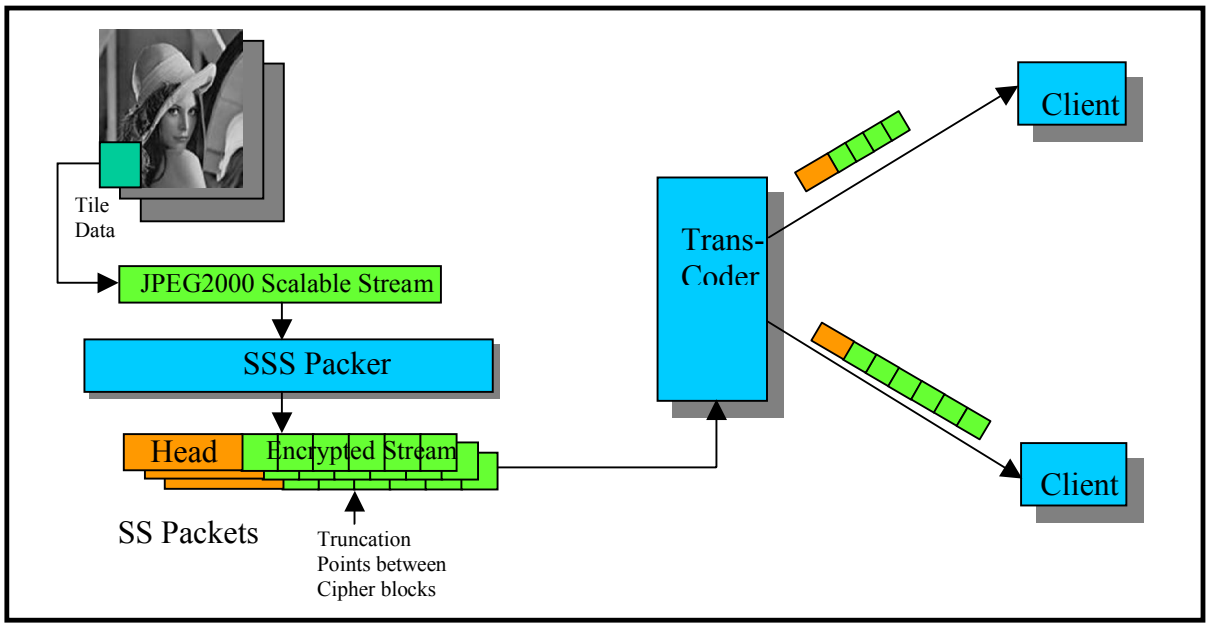

Figure 5: Secure Scalable Streaming System Overview

Progressive encryption of the compressed data is accomplished using block ciphers. In [12], experimental performance results were obtained using AES and 3DES block ciphers in cipher block chaining mode. In this mode, the first block is encrypted independently but all blocks thereafter depend on the previous block for decryption. Thus, if an encrypted stream is truncated at some point, the data that remains is not cryptographically dependent on any information in the truncated portion.

The un-encrypted header contains truncation point suggestions. As each SSS packet transits the network, intermediate transcoders do not need to decrypt the packet contents in order to perform transcoding operations. Rather, they refer to the un-encrypted header packet to locate trunctation points that are suitable to meet the bandwidth or quality requirement.

When the stream containing SSS packets arrives, the client strips away the encryption and reconstructs the video frame or tile using as much of the data as was received.

3.3.2 Adaptive Rich Media Secure (ARMS)

The authors of this work [23] seek a more generic, standardized approach than that taken in [11] and [12]. They attempt to accomplish this by extending the Internet Streaming Media Alliance (ISMA) security standard in a way that allows streams to be switched to meet network bandwidth or client device capabilities.

The innovation introduced by [23] is shown in Figure 6 below (based on Figure 4 of [23]). Instead of coding the multimedia object once using a fixed set of parameters (for example, a certain quality, resolution, etc.), the object is passed through a bank of MPEG coders that produce many copies of the same object with different quality and resolution settings. Then, each copy is streamed using the ISMA standard to an "ARMS Adaptive Streamer." This device observes and senses client processing capabilities and available network bandwidth and, using this information, it selects an appropriate stream to send to the client. In the event that the client processing capabilities or network conditions change, its task is to dynamically switch the stream being unicasted to the client.

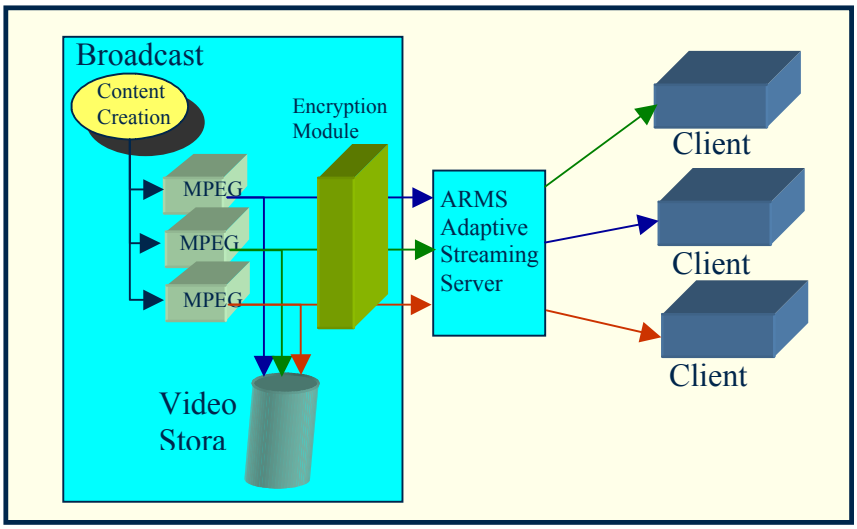

Figure 6: ARMS Architecture (based on Figure 4 of [23])

Stream switching is where this work offers innovation to the ISMA standard. The problem is to ensure that a client switching from one stream to another at the end of a "section" of content has the necessary elements to allow it to begin decrypting the next section in the new stream immediately.

The authors indicate that the ISMA standard requires the use of a stream cipher with random access capability (in order to enable resynchronization in the event of packet loss). This capability is provided in [23] by an encryption key and a counter based index parameter. A 128 bit counter value is encrypted using AES which is then XORed with 128 bit blocks of the streaming content. Two requirements for the index exist: a.) It must increase with successive sections of content; and b.) It must not be used twice with the same stream encryption key. This is not a problem if a single stream is assumed

(which is the case for ISMA). The index for the current section of content to be encrypted starts off where the last section ended and the key is changed when the index cycles.

However, ARMS employs multiple streams. To switch from one stream to the next, the client needs the key for the stream – which is potentially an easy problem if all the streams use the same key. The client also needs to know the starting index value for the "section" of content that it joins (stream switching occurs at "section" boundaries). Unfortunately, sections of content from different streams are likely to vary in size because they may have more or less information to convey in any given time period depending on the compression values chosen. The innovations to ISMA in [23] ensure that each section of content start with the same index value to ensure that switching can occur. For streams with different keys, the starting index value is chosen as the next index value available from the largest section in the previous time slot. In the case where the same key is used for all streams, the content to be encrypted from each stream is serialized within each time slot and the encryption index is computed for these as if they were a single stream.

3.3.3 Selective Encryption Methods

The authors of this paper [1] propose four fast and "lightweight" MPEG video encryption algorithms. The authors assert that, because of size and real-time processing requirements for multimedia content, confidentiality services using typical symmetric key encryption (such as DES, 3DES, AES, etc) may be both unwieldy and, in fact, unnecessary. The argument is that the confidentiality of the data needs only "provide [a] sufficient security level such that the cost to break the security system is much more expensive than to buy the data." ([1] p. 58)

The authors provide an excellent review of related work. Some of the key techniques that they summarize are listed below:

- Encrypt only MPEG video headers in a video stream.
- Encrypt I frames only.
- Randomly permute the way in which the DCT coefficients are converted from the 8X8 matrix to linear vector prior to encoding (the zig-zag pattern is randomized).

A more comprehensive review of selective encryption techniques is given in [19]. We will examine some of the results of that work shortly.

The authors provide four algorithms for performing lightweight MPEG video encryption. These are summarized from [1] below:

a.) Algorithm I - (no name given): the aim of this algorithm is to save encryption computation time. Essentially, the algorithm employs a secret key function that is used to map the values in the MPEG-1 Huffman codeword list to a permuted codeword list as

shown in Figure 7. There is a requirement in the algorithm to ensure that the permuted codewords are the same size as the original codewords or the stream compression will be sub-optimal. In order to recover the data properly, codewords are mapped back to the Huffman table using a key function (known only to the sender and receiver). The authors note that while no overhead is added to the MPEG codec, the "keyspace" is limited to the Huffman Coding table.

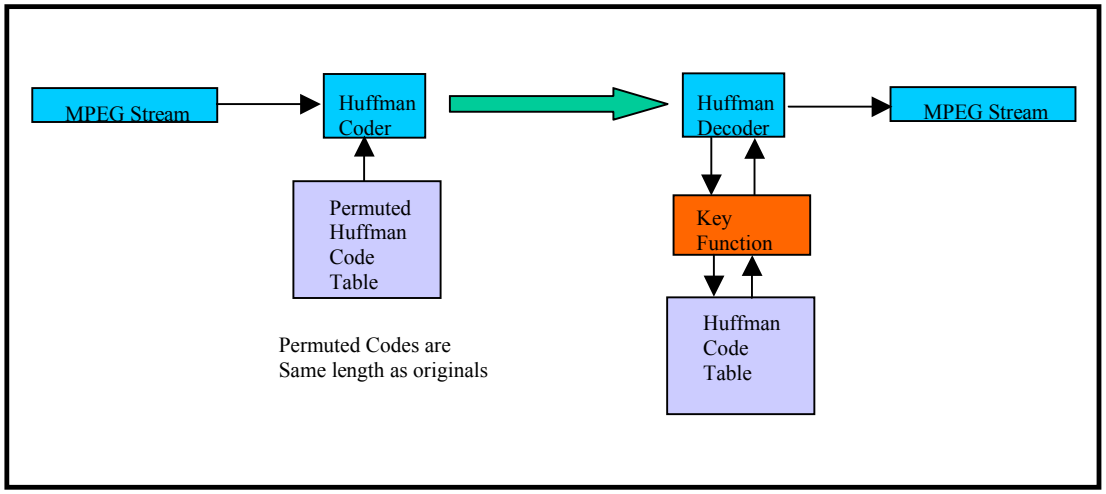

Figure 7: Algorithm I – Permuted Huffman Table

b.) Algorithm II – Video Encryption Algorithm (VEA): Recall that when the DCT is normally performed on an 8X8 block of unsigned integer data, the DCT produces coefficients that are signed. This algorithm targets the sign bits of the AC and DC coefficients in each DCT block. The encryption is performed by generating an m-bit key and using this key to perform a bit-wise exclusive-or (XOR) with each of the coefficient sign bits.

The security of this approach, as in the last approach, relies on the inverse DCT process. Because there are so few DCT coefficients, the reconstruction of the original data matrix from coefficients with the wrong sign will result in garbage. The apparent strength of the system depends on the key length. However, the algorithm is very weak to plaintext attacks. Finally, the encryption process is efficient because only a small percentage of the total data stream is modified.

c.) Algorithm III – Modified VEA (MVEA): The previous two algorithms focused exclusively on I-frames. In the MVEA algorithm, B and P frames are also targeted for encryption. In this case, only the sign bits of the DC coefficients in I-frames were encrypted since it reduced the computational burden and was still effective given other encryption measures. For B and P frames, the sign bit of the motion vectors were also encrypted. The video decoder uses the motion vector to reconstruct a portion of the current frame from a reference frame. Changing the sign bit of the motion vector

changes the reference direction and the MB is rendered incorrectly. As with VEA, MVEA is weak towards plaintext attacks.

d.) Algorithm IV – Robust VEA (RVEA): The aim of RVEA is to provide a lightweight encryption scheme that is robust to plaintext cryptographic attacks. The general concept is (as in MVEA) to encrypt only the sign bits of DCT coefficients in I frames and motion vectors in P and B frames. In this way, only a small percentage of the total bit stream needs to be encrypted and some computational savings are realized. However, instead of performing encryption using an m-bit key and the XOR function on the sign bits, a "heavyweight" symmetric key encryption algorithm (for example DES) is used. This renders the encrypted sign bits next to impervious to plaintext attacks. The authors report that sign bits represented less than 10% of the total bit stream and a 90% savings in computation time was observed (compared to encrypting the whole stream).

3.3.4 Discussion

In [12], the authors provide confidentiality to their streaming data using symmetric key encryption on the total payload. They attempt to provide fine-grained scalability to the video stream without decrypting or transcoding it (the operations are performed in the "secure domain"). The authors claim an average overhead (compared to end-to-end encrypted delivery) of approximately 2-2.5%. However, SSS transcoders are required to be present in the data stream as are SSS aware clients.

The approach taken by [23] is similar in that a full implementation of symmetric key encryption is used on the payload. However, scalability is addressed specifically by allowing the client to switch to a lower resolution stream in the event that transport capabilities change. While no data is provided to indicate what overhead is introduced by this scheme, the use of ISMA implies a certain packet overhead for meta data. There is also a requirement to employ ISMA aware clients.

The authors of [1] explored confidentiality requirements through selective encryption of portions of the payload from a video stream. The authors of [1] obtained very good results in terms of "scrambling" content. VEA required only 1.8% overhead in processing time while the total time spent encrypting during compression while using RVEA was measured at approximately 2.5%. The approaches outlined in this work do not involve packet overhead nor do they require handlers in the traffic stream.

Much work appears to have been done in selective encryption methods. The authors of [19] reviewed over seventeen works targeted at selectively encrypting various image and video bitstreams. They conclude that, while the work presented is promising, there are several shortcomings which will require further effort. One of these perceived shortcomings is the lack of an integrated key management system: "it cannot be separated from the design of secure multimedia distribution." ([19] p. 7]) We will look at key management schemes next.

3.4 Key Management

3.4.1 Introduction

In the previous section, we discussed video streaming. It was noted that encryption appears to be the dominant approach to securing the confidentiality of streaming applications. However, in almost all cases where encryption is used, a means to securely distribute and periodically update encryption keys is needed. In the literature, this is frequently assumed out of scope.

In this section, we discuss aspects of key distribution and update – elements of *key management*. We limit the context to broadcast or multicast conditional access applications. Such applications include pay-TV, pay-per-view, network news, online gaming, and multimedia conferencing [4][9].

3.4.2 Characteristics

There are several desirable characteristics of a key management system. These are drawn from [13][4] and [8]:

- Key management must be able to efficiently deal with a possibly high turnover rate in end-users. This requirement has two constraints:
    - End-users who join a session[1] must not be able to access media that has already been multicast; and
    - End-users who leave a session must not be able to access multicast media from the time of departure onwards.
- Key management should have minimal impact on the communication and computation requirements of end-user access devices;
- Key management should be *scalable* in the sense that incremental increases in the number of end users in any session should not cause unmanageable performance degradation.
- Key management should be resilient in the face of a key compromise in the sense that detection and recovery should be transparent to the end-user (added by us).

[1] A "session" is loosely defined here. Depending on the application, the session may be a channel or stream or an individual program or multimedia event (i.e., a teleconference). In some circumstances, the end-user may expect and be entitled to access to a "program" or "event" from the start even though he/she has "tuned in" late.

Public key algorithms could be used to address key distribution. These algorithms neatly solve the problem of mutual authentication. Algorithms such as Diffie-Hellman key exchange could be used to securely derive a mutual key between server and client. However, while authentication need only be performed once between a client and server (when the client joins a session), key exchange may be required quite often. This is problematic since public key algorithms tend to be slow and require substantial client side processing. Therefore, most key distribution models do not make use of public key cryptography for key exchange.

3.4.3 Flat Key Distribution Model

In this section we discuss a flat key distribution model (as described on page 545 of [13]) in which a central server broadcasts/multicasts one or more multimedia sessions to a set of end-users. This architecture is referred to in [27] as the "Star Topology." We make reference to both in constructing the description in this section.

Consider figure 8 below (based on Figure 1 of [27]). At time t, the server has a media key ($M_{K(t)}$) used to encrypt the multimedia content being streamed or otherwise served. There are three end-users E_1, E_2, and E_3. Each end user has a secret symmetric key K_{E1}, K_{E2}, and K_{E3} used to secure communications between itself and the server for the purposes of performing key update operations.

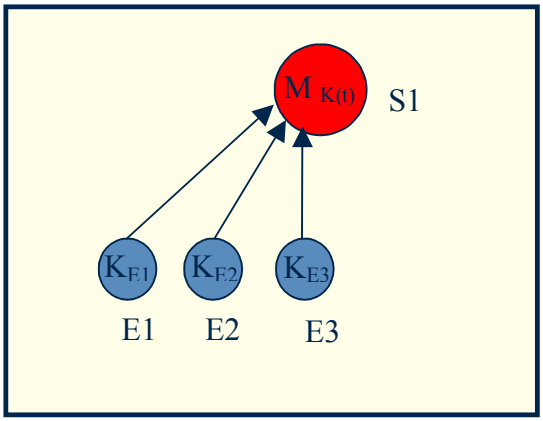

Figure 8 (based on Figure 1 of [27])

We may now consider three cases with the model of Figure 8:

Case 1: A new user, E_4, joins the multicast. The following operations occur:

- E_4 requests to join the multicast.
- The server and E_4 perform mutual authentication via some means. As part of this authentication, the Server and E_4 negotiate a symmetric key K_{E4}.
- The server generates a new media key $M_{K(t+1)}$.
- The server encrypts the new media key with the old media key and multicasts this to E_1, E_2, and E_3.
- The server encrypts the new media key with the symmetric key K_{E4} and sends this to E_4 (but only after E_1, E_2, and E_3 have acknowledged receipt of the new media key.

In this case, the new user is prevented from being able to view the content that has already been streamed.

Case 2: An existing user E2 leaves the multicast. The following operations occur:

- E_2 requests to leave the multicast.
- The server generates a new media key $M_{K(t+2)}$.
- The server encrypts this new media key with K_{E1}, K_{E3} and K_{E4} and sends each to the appropriate end user.

In this case, the leaving user no longer has access to the media content.

Case 3: Periodic re-generation of the Media Key.

Although this case is not specifically discussed in [27] or [13] for the flat key distribution model, we may easily intuit the operations required for a Media Key update. Such an update is generally required to ensure the continued security of the system against cryptanalysis attacks. The following operations would be required:

- The server generates a new media key $M_{K(t)}$.
- The server encrypts this new media key with the symmetric keys of all existing users and then sends each to the appropriate end user.

While the architecture of Figure 8 is simplistic, it is one that could naturally evolve (and probably has). However, we find that the server must perform (n-1) symmetric encryptions (where n is the number of users at time t) and send (n-1) messages in Case 2. Case 3 has the same complexity. The server is also required to retain (n+1) symmetric keys. As indicated in [27] and [13], the complexity of the server communication, computation, and storage requirements is linear making this approach decidedly non-scalable. We next discuss an improvement on this basic (but intuitive) architecture.

3.4.4 Key Graph Model

In the key graph model of [27], scalability is achieved by the introduction of subgroups. In this approach, the server creates a key graph with two types of nodes: u-nodes that represent users and k-nodes that represent keys. Each user has a key set whose elements are the keys found on the directed path between the user's u-node and the root node. Figure 9 (based on Figure 1 of [27]) below shows a key graph with one layer of subgroup keys.

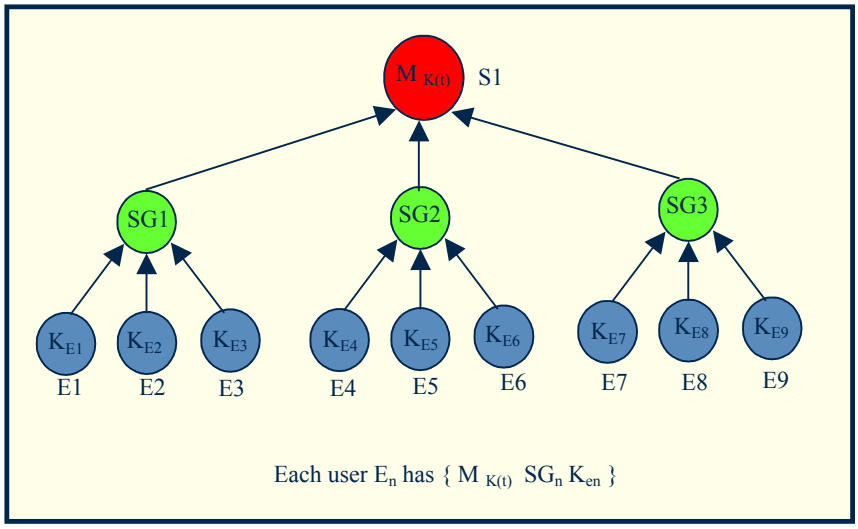

Figure 9 (based on Figure 1 of [27]): Key Graph Model

In the figure, each end-user has three keys: a Media Key $M_{K(t)}$ (which is referred to in [27] as the Group Key – we shall preserve our notation here), its own encryption key E_{Kn}, and a subgroup key SG_n. For example, end-user E4 has E_{K4}, SG_2, and $M_{K(t)}$. In the flat model discussed earlier, end-users did not have sub-group keys.

The chief advantage conferred by the division of end-users into groups is a savings in server processing. Let us reconsider Case 2 (where an end-user leaves) and identify the leaving user as E9 in Figure 9. As in the flat model, the server must generate a new Media Key $M_{K(t+1)}$ and distribute this to all remaining users. For the first and second group, the server needs only perform two encryptions ($M_{K(t+1)}$ using SG_1 and SG_2) and unicast/broadcast the results to users in these two groups. In the flat model, the server needed to perform six encryptions of $M_{K(t+1)}$ (one for each user).

However, in the key graph model, the server must also change SG_3 since E_9 had a copy of this key. The server generate $SG_{3(t+1)}$ and encrypts it using K_{E7} and K_{E8}. Finally, the server must give E7 and E8 the new media key $M_{K(t+1)}$ so it encrypts it using the new group key $S_{G3(t+1)}$.

The total number of encryptions in the example of Figure 9 is five (versus eight for the flat model). As [27] indicates, for large numbers of users organized in a full and

balanced tree, the savings in processing time at the server can be substantial. Considerably more savings is realized when one considers Case 3 – periodic re-keying. Instead of nine encryptions, the server needs only perform three. This could be advantageous where the requirement for re-keying is frequent.

3.4.5 Secret Sharing

An interesting approach to the key management problem in conditional access systems has been proposed in [7]. This work recommends a key transport protocol based on a threshold scheme in which a secret S is broken up into t "shares." To reconstruct S, it is necessary to be in possession of a certain number (or "threshold") of the t shares.

In [7], the authors propose pre-positioning one or more of the t shares on the end-user device (say, the smart card in a set-top pay-per-view system). The concept is then to send one or more of the remaining t shares over the communication channel (possibly in cleartext) with the encrypted media. The end-user device would compute the secret based on all t shares and decrypt the incoming content.

The work in [7] uses Shamir's [53] thresholding scheme in which:

> "the secret S is coefficient a_0 of a random (t-1) degree polynomial
>
> $$f(x)=(a_{t-1}x^{t-1}+\ldots+a_1x+a_0) \pmod p \quad (1)$$
>
> over the finite Galois Field GF(p), where p is a prime number larger than both S and n. Each of the n shares (x_i,y_i) is a point on the curve defined by the polynomial f(x)… As a polynomial of degree (t-1) can be uniquely determined by t points, the secret can be computed from t shares." ([7] p. 5])

A trivial example of using a first degree polynomial (of the form $f(x)=a_1x+a_0$) is now given to illustrate this proposal (based on a similar example included in [53]). A first degree polynomial describes a line. To completely specify the line, two points are needed. As part of the distribution system, one of these points (x_0,y_0) is pre-positioned with the end-user(s). To stream encrypt or scramble data from the server to the end-user(s), the server first computes a secret S as a point on the y-axis (0,S) and uses this to scramble or encrypt the content. It then constructs a polynomial (a line) that passes through both (0,S) and (x_0,y_0). Finally, it chooses a third point (x_1, y_1) on the line and sends this ahead of the scrambled data. This is called the "activating share." The end-user(s) receive (x_1,y_1) and combine it with (x_0,y_0) to solve the system of equations and determine a_0 (which is (0,S)). This value is used to decrypt or de-scramble the data stream.

Note that the server can repeat the process to generate a new secret while consistently using the pre-positioned secret at the end-user. The authors of [7] note that he security of

this system demands polynomials of higher order. For each polynomial of degree (t-1), there are (t-1) points pre-positioned at each end-user. The "activating share" is that last point on the function needed by the end-user that will reveal a_0. Computation of a new key requires the server to fix (0,S) and determine (x_1,y_1) from a function that passes through all of the points that make up the pre-positioned information.

3.4.6 Centralized Key Management with Secret Sharing (CKMSS)

In [5] and [8] the authors have merged the work in [7] with the concept of key graphs in [27] and label the new system Centralized Key Management with Secret Sharing (CKMSS). In addition, they attempt to show how their implementation could be used with scalable video.

The important aspect of the Secret Sharing approach in [5] is that, using the same set of pre-positioned shares, the server can change the secret key generated by the end-user simply by sending a different activating share (recall that the activating share used in this work is the last point that each client needs to uniquely define the polynomial and therefore determine a_0 - the secret).

In the CKMSS scheme, the authors have essentially taken the key graph model of [27] and, wherever a key would normally be distributed to a key node (for example, a sub group key node or and end user), *a set of shares is distributed instead*. Both the group keys and the client keys can then be generated locally by the client using an activating share sent by the server. Note, however, that the shares must be sent in a protected manner (i.e., encrypted). This is different from [7] in that the secret shares in that work were assumed to be pre-positioned at system production or rollout.

Let us examine how a member leaving a sub-group would be handled by CKMSS. Figure 10 shows user E9 leaving the session.

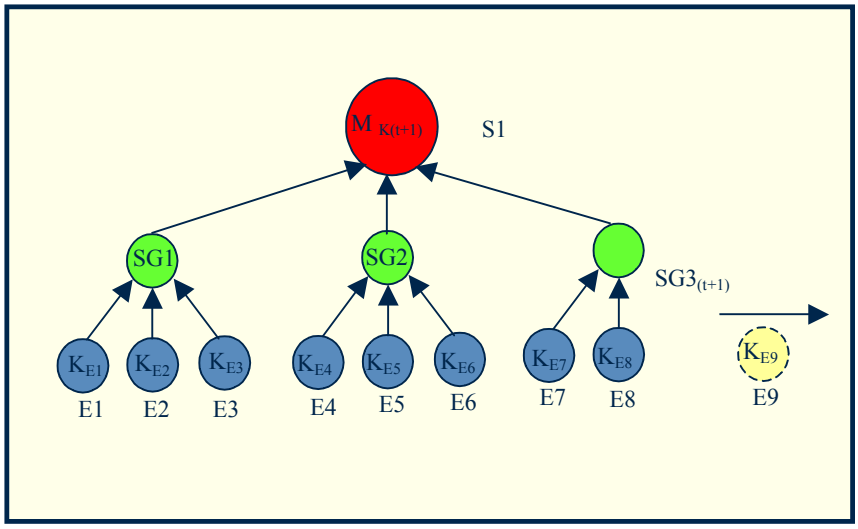

Figure 10: E9 Leaves

a.) **Server Computation**: Assume member E_9 of Figure 10 leaves the tree. E_9 belongs to sub-group SG3. Therefore, in order to "clean up" behind E_9, the server needs to issue a new SG3 key and distribute this to E_7 and E_8 (because E_9 knows the sub-group key). The server also needs to generate $M_{K(t+1)}$ (because E_9 also knows that). In [5], $M_{K(t+1)}$ is chosen and set of (t-1) shares for this key are computed. The server also chooses $SG3_{(t+1)}$ and computes a corresponding set of (t-1) shares. The server then encrypts the shares for $M_{K(t+1)}$ with the sub-group keys for SG1, SG2, and $SG3_{(t+1)}$. It also encrypts the shares for $SG3_{(t+1)}$ with the individual keys of E_7 and E_8. Finally, it multicasts all of the encrypted items *along with the activating share* for the new $M_{K(t+1)}$ and $S_{G(t+1)}$ keys.

b.) **End-User Computation**: Each end-user decrypts the (t-1) shares for $M_{K(t+1)}$ while E_7 and E_8 also decrypt the shares for $SG3_{(t+1)}$. Each client then uses the activating share it received to construct the Media Key $M_{K(t+1)}$ E_7 and E_8 also use the activating share for the new group key to construct $SG3_{(t+1)}$.

While this process may appear to add undue complexity at the end-user side (solving the system of equations to determine new keys), the authors of [5] indicate several benefits after performing a number simulation results. The chief advantage is obvious without any simulation at all: periodic re-keying requires only a single multicast of a fixed length activating share and this share does not need any encryption. The approach in [27] requires at least *n*SG encryption operations (where nSG is the total number of sub-group key nodes). In systems where periodic re-keying is required frequently, this aspect may confer some advantage. Some of the other conclusions provided in [5] include:

- Increasing the number of shares per node incurs a (mildly) non-linear computation cost. The authors note that the cost could be justified by matching it against the threat and multimedia asset value; and
- The processing time per request increases linearly with the logarithm of the group size.

As a final note, the CKMSS model presented in [5] was also used to secure scalable video which consisted of a base layer and one or more enhancement layers. Each layer was multicast independently. The approach was reasonably straightforward. Each layer was assigned a separate Media Key MK. However, only the base layer MK was changed after a join or leave since the enhancement layers – in the absence of the base layer – did not convey usable information. This work was also reported in [20].

4 Watermarking

4.1 Introduction

Watermarking is seen as a technique, complementary to encryption, for the secure storage and distribution of multimedia content [3]. In a watermarking system, information is inserted directly into a multimedia object at the expense of a hopefully imperceptible degradation in the quality of that object.

Digital watermarking has obvious parallels with steganography [32][34][31] and physical paper watermarking. Steganography is a Greek word meaning covert communication and is a sub-discipline of data hiding. In steganography, data is hidden in some "cover object" such that the object itself attracts no particular attention (except by those who wish to communicate securely). In recent years, a host of electronic tools have become available for embedding secret information into digital images, video, and audio (a simple "Google" on "steganography tools" will receive thousands of hits). Such tools are useful in an internet perspective because they can help obscure communicating parties; as opposed to sending a multimedia object directly (perhaps by mail – which automatically reveals the communicating parties), the object can be downloaded during what looks like a normal browsing session from just about anywhere. If the web server is very active, it will be difficult to determine communicating parties from the "noise."

While steganographic messages in cover objects must remain hidden, the principle concept of a paper watermark (which appeared during the era of hand-made papermaking about 700 years ago [32]) is that the embedded information is visible. This information was used by artisans and merchants to authenticate certain characteristics of the paper being purchased. In some countries, currency is watermarked to foil counterfeiting operations. In the sections that follow, we shall see the parallels between digital watermarking technologies.

4.2 Uses of Watermarks

Watermarking technologies can be used for the "identification of the origin of content, tracing illegally distributed copies and disabling unauthorized access to content" ([6] p.243). *Origin of content* has the requirement that an author's proof of ownership can be unambiguously determined in cases of dispute [29]. Digital watermarks are seen as a technique having the potential to fulfill this requirement. The mechanics of this system operate as follows. A digital watermark is created and inserted into the media which is then circulated according some distribution model. At some later date, the author may observe his/her work being used in a way that was not intended or authorized. The author can commence civil or administrative action to re-assert control of the work or have payment rendered. As pointed out in [29], the watermark must be resilient to attacks by pirates who may have near infinite time and resource to attack a copyrighted object.

A second use of watermarks is to identify *copyright traitors*. [29][30][55]. Watermarks used in this context are often referred to as *fingerprints* because each end entity receiving a copy of a media file will have an entity-specific identifier embedded in the media file as a watermark. Then, if the end-entity illegally distributes their copy, the authors have a way of determining who they are and taking appropriate action. There are still many legal and research issues involved in this use of watermarking.

4.3 Properties of Watermarks

The following are the properties of watermarks drawn from reviews of watermarking and data hiding ([31],[32],[34],[6],[54]). Watermarks can have one or more of these properties although several properties are exclusive of other properties.

- Robust: there are two categories of "attack" that can be perpetrated against a watermarked object. The first category includes un-intentional attacks that involve all of the standard data processing operations that one might expect. For example, in the case of image or video, processing operations include potentially lossy compression, image cropping, zooming, enhancement, etc. The other category of attack includes those operations perpetrated "with the purpose of impairing, destroying, or removing the embedded watermarks"([34] p. 1081). A truly robust watermark, then, is one that can remain intact through un-intentional attacks and at the same time render obvious intentional attacks. Making an attack obvious usually implies that the watermarked object is damaged or destroyed in some perceptually obvious way [32].

- Fragile: a watermark with this property is "destroyed as soon as the object is modified too much" ([32] p. 1063). This type of watermark is designed specifically *not* to be robust to modifications and may be used during integrity verification of an object. A distinction between fragile and semi-fragile can be made [31]: a truly *fragile* watermark will be destroyed under the slightest modification whereas a *semi-fragile* watermark will only be destroyed after a certain threshold in modifications has been reached.

- Imperceptible: as indicated in [34 p. 1082], "the design of a watermarking method always involves a tradeoff between imperceptibility and robustness." Embedding a watermark into media should not introduce any perceptible change to the media; it should be imperceptible. However, robustness requires that the watermark be embedded in "the most perceptually significant components" ([32] p. 1064) to resist intentional attacks without crossing the threshold of actual perception by a human. As [34] indicates, this threshold of perception is a subjective one and normally requires human input to determine.

- Visible: While most of the literature has focused on watermarks having the "imperceptible" property [32][31], there are occasions where a digital watermark

may be required to be visible to the user. Such occasions are strongly associated with traditional paper watermarking (logos, copyright notices, etc).

- Un-ambiguous: A watermark having this property should be able to identify a single author or content owner and should degrade gracefully under attack [31]. This property targets a class of attack whose scope is to introduce new (or modify existing) watermarks to make the content appear to be owned by several entities thereby throwing ambiguity into the question of copyright.

4.4 Watermarking Process Model

In order to fully understand the elements and processes involved in watermarking, we now examine the watermarking model described in [29] and re-presented/paraphrased here. The essential constituents of an end-to-end watermarking system are shown in figure 11 (figure 1 from [29]).

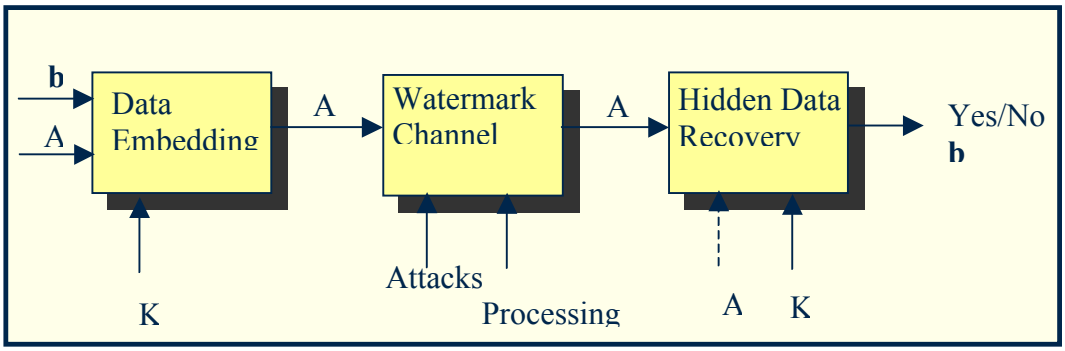

Figure 11: Watermarking Model (figure 1 from [29])

4.4.1 Data Embedding

The role of data embedding is to mix the watermark information with the media to be watermarked (represented as **b** and **A** in Figure 11, respectively). Depending on the implementation, the embedding module may also incorporate the use of a secret (symmetric or asymmetric) key, **K**, as shown. This key is used with some encryption algorithm in order to scramble the watermark prior to embedding it into the cover object. Encrypting the watermark may be required if there is a need to keep the watermark information confidential and if the watermark recovery mechanism is widely known. Another obvious benefit to encrypting the watermark is non-repudiation. That is, it is not possible to forge someone else's watermark in an object (unless access has been gained to their secret keys).

The actual embedding process appears to be divided into two broad categories as described below:

a.) Blind (Public, Oblivious) Watermarking: In a system of blind embedding, the creation of the watermarked media has no dependency on the information in the original unmarked media. This has important performance and accuracy implications for watermark detection. Spread spectrum techniques represent an important class of blind embedding [35] but are not necessarily constrained to this category. Blind embedding is also known as *public marking* [32].

b.) Informed (Private, Non-Oblivious) Watermarking: This system of watermarking makes use of the original media (the "cover signal") to assist in the recovery of the watermark. As indicated in [35], early blind embedding systems considered the cover signal to be noise. However, it was realized that the cover signal could (in general) also be available at the decoder as side information (i.e., not transmitted over the watermark channel and therefore having no bandwidth implications) and could therefore be used in recovering the watermark. In this context, the technique has become known generally as "informed watermarking" (but also goes by "known host state", "dirty-paper coding," and "writing on dirty paper (WoDP)" [35]).

4.4.2 Watermark Channel

The watermark channel describes the possible transformation of $\mathbf{A}_w$ to **A'** that may occur as a result of either routine image processing or intentional attack. We will revisit the topic of watermark attacks in more detail shortly.

4.4.3 Watermark Detection or Recovery

The data recovery process is designed to perform one of two operations. It either detects the presence of a watermark and asserts a boolean flag accordingly or it reads and extracts the watermark. In the case of simply *detecting* the presence of a watermark (common with spread spectrum techniques), the original watermark information (**b**) is generally required whereas the original, unmarked cover object is not. In the case where a watermark is to be recovered, the original cover object (**A**) is required if the insertion process was informed. A decryption key **K** is also required one was used during the insertion process.

4.5 A Review of Watermark Attacks

4.5.1 Introduction

There are a number of general attacks against watermarks [31][33]. Most of these seek to challenge the robustness of watermarking schemes from the point of view of normal or "expected" data processing as opposed to being deliberate attempts to foil the watermark. For example, [33] describes StirMark, a tool designed for the basic testing of image robustness. Some of the manipulations possible with StirMark include low pass filtering, colour quantization, compression, scaling, cropping, rotation, and random geometric distortions (such as image stretching, shearing, shifting, and bending). Some of these operations – either alone or in combination – are quite successful at disabling watermarking schemes (for example rotation and scaling against correlation based watermark detection/extraction [31]).

In the following sections, we describe a number of deliberate attacks against various watermarking schemes. We arrange the sections according to the taxonomy of watermarks introduced in [56] which include authorship (or ownership), fingerprint, and validation (or integrity) watermarks. Although that taxonomy is applied against software watermarks, the groupings are non-the-less useful.

4.5.2 Protocol Attacks on Authorship Watermarks

Unique approaches to addressing *protocol* attacks are presented in [36] and [37]. An attacker can violate an author's proof of ownership if he/she can remove a watermark from an object. This is what the property of robustness under attack seeks to prevent and has been the focus of substantial research. However, "many of [the] existing schemes have not addressed the *ends* of invisible watermarking schemes" (emphasis theirs)[1] ([36] p. 574). That is, to be useful, an embedded watermark needs to be unambiguous in a court of law. The authors of [37] assert that the process may be "subverted entirely without removing any watermark contained in the multimedia objects" ([37] p. 111) by attacking the dispute resolution process as opposed to the watermark itself. Thus, a protocol attack seeks to introduce uncertainty and ambiguity as to the rightful owner of a multimedia object only when that object becomes the source of a dispute. As [38] explains, "one of the primary problems to be addressed by watermarking methods is their ability to make a counter-claim [against a watermarked work] *practically impossible*" ([38] p. 470).

The authors of [37] divide protocol attacks up into two classes:

[1] This paper was published in 1998. This statement may not be current.

- Copy attacks: In this class of attack, the objective is to copy an existing watermark from one object to another without knowledge of the watermark or any key that was used in conjunction with the watermark.

- Ambiguity attack: This class of attack has been referred to in the literature by different names (e.g., "finding ghosts" [57]) yet the objective is consistent; the general case involves "computing a watermark that was never inserted in an object… but still can be detected there" ([37 p. 113). As an example of this type of attack we next contemplate in detail the Single Watermarked Image Counterfeit Original (SWICO) attack described in [6] and paraphrased below:

  > The SWICO attack is an example of an inversion attack (which is identified in [37] as a special case of ambiguity attack). Let us assume the canonical security entities Alice and her adversary Eve (one may wonder if Eve is always after Alice as a result of a failed love triangle involving Bob). Eve, for whatever nefarious purpose, would like not only to lay claim on a multimedia object, M, already watermarked by Alice (and represented here as M_A), but she would like to eliminate any possibility of a court determining that the object M originally belonged to Alice.
  >
  > One way to attempt this is for Eve to obtain Alice's object M_A and introduce her own watermark into it (which should be possible if the object has the robustness property). However, the problem with Eve's scheme is that Alice will have the original, M, that does not contain Eve's watermark. If she presents this to the courts, Eve will go to jail (or at least be fined a very large sum). Eve must find a way of inserting her watermark into Alice's original but without getting access to it. In that case – and that case alone – a courts will not be able to rule in Alice's favour.
  >
  > How does Eve accomplish this? First, we assume that Eve has complete freedom to choose her watermark signal and watermark detection algorithm[2]. According to [36], she can take the watermarked object M_A and identify within it a set of features to use as her watermark, W_E. She then designs a watermark detector with a decision statistic, S_E, tailored to find her "phantom" watermark in M_A. Finally, she removes W_E from M_A and calls this object, M_{AE}, her "original" (if M_A is robust, there should be no appreciable degradation in quality as a result of this operation).
  >
  > One of the crucial factors here is designing an appropriate S_E. If this is done correctly and meets the constraints outlined in [36], then when Alice's original object M is applied to Eve's watermark detector, the

[2] Eve may be constrained to using the watermark detection algorithm that Alice used. However, because detection is statistical in nature, there exists a probability of false alarms (i.e., detecting a watermark that was not inserted). Using this fact, and depending on this probability, it may still be possible to find a suitable watermark in the target object [37].

detector will report the presence of Eve's watermark (recall that M and M_A are not significantly different so that a judicious selection of S_E in M_A will likely also be found in M).

The result is that Eve has a (counterfeited) object containing Alice's watermark. Alice has an original object that contains Eve's contrived watermark. And, in the distribution system, there are objects containing both watermarks. It is therefore not possible for an arbitration body to determine rightful ownership. Note that this attack is only applicable against watermarking schemes that the authors of [36] call *invertible*. A scheme is invertible if it is possible to invert "a watermarking encoding function ε, in order to "remove" a watermark from an image rather than insert one" ([36 p. 579) (as was performed by Eve to subtract W_E from M_A).

The work in [38] attempts to eliminate protocol attacks by proposing a watermarking protocol that limits the degrees of freedom allowed when choosing watermarking components. It is, in fact, "a list of restrictions to be placed on watermarking methods so that they meet the end requirement, viz. unambiguous resolution of ownership" ([38] p. 477). They also note that time-stamping has been proposed as a solution to address the ambiguity problem. However, while that approach has merit, the requirement for a trusted time source in high availability mode could be seen as a limiting factor.

The work in [37] attempts to eliminate or limit some protocol attacks by incorporating the use of cryptographic signatures in developing the watermark. In this case, there exists a trusted third party who generates digital signatures that are used in the watermarking process to attach a validity constraint to possible watermarks. Thus, not only does a watermark need to be detectable within the object (this set includes the actual watermark and all "false positives"), but it also must be valid. The strength of the mechanism relies on the inability of an attacker to forge a watermark that is both present in the object and valid in a reasonable time.

4.5.3 Attacks on Fingerprint Watermarks

4.5.3.1 Collusion Attacks

Collusion attacks are unique to fingerprint watermarks because a different watermark is embedded with each copy of the media object. The general mechanism of a collusion attack [29][30] is as follows. A group of users brings together their individually watermarked copies of the same media object. These are then compared or averaged to produce a single copy that has no watermark or a watermark that no longer is within the detection threshold for the target decoder.

However, [29] indicates that this type of attack only works with blind watermarking – where the embedded watermark is independent of the information in the watermarked media. Therefore, it would seem that an obvious solution is to simply make the

watermarking scheme informed. However, the consensus appears to be that the "design and optimization of SS [spread spectrum] systems seems to be easier than the design and optimization of binning schemes" ([35] p. 55). In other words, the work factor and associated performance and complexity costs involved in switching from blind schemes to informed schemes would have to be carefully evaluated against the costs incurred by collusion attacks.

4.5.3.2 Protocol Attacks

The authors of [29] indicate a straightforward protocol attack against fingerprinting techniques: "a buyer whose watermark is found in an unauthorized copy cannot be prosecuted legally since he can claim that the unauthorized copy was created and distributed by the seller" ([29] p. 34). In this case, the author or distributor generates and embeds the unique fingerprint for each consumer at the source and this opens the door the protocol attack.

One solution to this problem is to implement a trusted third party (TTP) whose function is to implement all watermarks in a secure and "chain-of-evidence" preserving manner [29]. However, it is quickly acknowledged that this entity could quickly become a bottleneck and may not be suitable for all DRM architectures.

Another solution is to perform receiver side watermarking. However, as pointed out in [30], there would be additional elements to any receiving device (tamperproof hardware, etc) that may not either be present or the cost of which may deter potential consumers from investing. Technologies for digital rights management may, at some future date, fill in this capability gap and make such end-user operations feasible.

The work in [30] represents a very interesting approach to the fingerprinting of multicast video. In this work, they propose a Joint Fingerprint and Decryption (JFD) architecture to assist in keeping track of pirated video content. In this scheme, a single encryption key, **KS** is used by a broadcaster to encrypt multimedia content. Each client is issued with his/her own unique decryption key, **KC**. While KS≠KC, they are correlated to a certain degree which provides a quantity of mutual information. Thus, the decryption performed by the client will not be perfect and this imperfection represents the unique fingerprint.

The tradeoff in this scheme is the amount of mutual information versus multimedia quality. Less mutual information leads to a larger watermark space – but at the risk of introducing noticeable distortion into the multimedia. While this work is vulnerable to both collusion and protocol attacks (the keys are generated for the client by the broadcaster), the focus is on tracing multimedia pirates in a *computationally efficient* manner and in this respect it appears to be somewhat successful. Encryption only needs to take place once at the source and decryption is inherently combined with watermarking. While information about possible pirates may not stand up in a court of law, it could "enable the seller to identify potentially deceitful customers and break off any further business relationship with them" ([29] p. 34]).

4.5.3 Attacks on Authentication/Integrity Watermarks

In this final section, we consider attacks against authentication/integrity watermarks. Recall that watermarks used for the purposes of integrity should have the property of being fragile – too many changes (or perhaps any at all) should result in the watermark being destroyed. In principle, this would guarantee that any modification of the cover object would instantly become obvious. Examples of where this type of watermarking would be useful include photographic evidence in a court of law and medical applications.

An excellent review of possible attacks against authentication/integrity watermarks is given in [58] and summarized here. The scope of the paper is limited to malicious attacks against invertible watermarking schemes. These are schemes in which the watermark can be removed from the multimedia object in such a way that the object is returned to its pre-watermarked state which is important if the image is to remain unchanged in the least.

The authors describe one means of doing this by selecting a set of perceptually insignificant elements and compressing these (in a lossless fashion) to make room for the watermark. The final image is then composed of the perceptual elements, a compressed set of elements, and a watermark. To reconstruct the original image, the watermark is removed and the compressed elements are decompressed into their original locations. The watermark is composed of a one-way hash of the original image file and encrypted using either a symmetric key or asymmetric keys.

The authors identify three types of attack that can occur against invertible watermarks which are paraphrased from [58] below:

- Key distribution attacks: this type of attack seeks to disassociate the key used to sign the cover object from the key used to verify its authenticity. This attack is only possible in the case where symmetric keys are used to encrypt the hash of the object because there is no binding between the signing key and an end user or device. A malicious person can re-sign the image using an arbitrary key and make it look like the object originated from him/her.

- Integrity verification attacks: this type of attack seeks to force the verification phase of an object to report a that verification was good when, in fact, it was not. In general, the attacker will know the verification keys associated with the object or person.

- Original reconstruction attacks: this type of attack seeks to reconstruct the original multimedia object (which should be kept confidential) from properties existing in the watermarked object.

5 Digital Rights Management

5.1 Introduction

Digital Rights Management (DRM) has been defined generally as a "systematic approach to copyright protection for digital media" [59]. As we shall see, DRM involves a whole host of technologies and processes – some elements of which have evolved to the point of utility and some of which are still to come. We shall also see that attempts to move forward with DRM have disturbed a hornet's nest of legal, privacy and ethical issues [60][61].

In the following sections, we define DRM by exploring current models of DRM systems. We then explore the approaches to DRM. Technical approaches constitute the bulk of the investigation. However, while legal approaches are given lighter treatment, they do deserve some attention as they have caused much of the controversy now troubling DRM.

5.2 Elements of a DRM System

A "typical" DRM model is described by [40]. We use this model as a starting point in the portrayal of the entities and interactions that one might expect in a DRM system. Figure 12 (Figure 2.1 from [40]) depicts this basic DRM model.

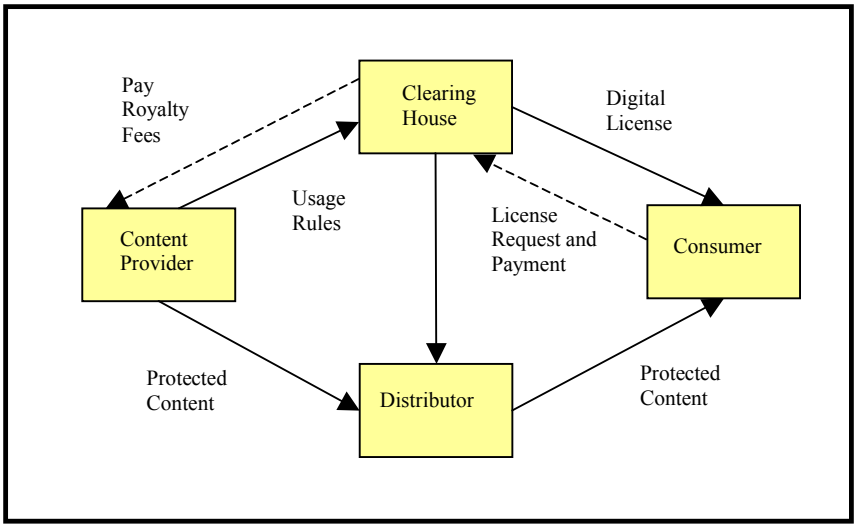

Figure 12 (figure 2.1 from [40]): DRM Model

The content **provider** and **consumer** form the core of any DRM system. The content **provider** is the originator of multimedia content. This could be a user, an e-book company, a movie studio, etc. This entity determines the "usage rules" or "rights" for the content that will eventually specify how the **consumer** is allowed to use the content. These rights will depend on the business or marketing model being used by the **provider**.

For example, if the **provider** was in the business of video sales, the rights package might contain constraints similar to "never allow this to be copied or transferred to other people" and "playable by this **consumer** without limit." If the movie was downloaded from an enterprise that makes its business from "renting" movies, it might also contain the constraint "playable only until next Saturday (after which the content is useless)."

The rights package is written or "expressed" in some language that can be un-ambiguously interpreted and enforced by the media player engine at the consumer's site. When associated with some multimedia content, the rights package can be referred to as a **license** for that content issued to the **consumer**.

After creating the content in a form suitable for distribution, the provider can "protect" it, perhaps using technical methods already discussed. For example, a video might be scalably compressed for video streaming, encrypted for confidentiality in storage, and watermarked for non-repudiation of origin. The provider then moves the content to a **distributor** for access by an audience of interested consumers. At the same time, the provider would send any decryption keys and the rights package for that content to a **clearing-house**. A transaction with these entities would involve the consumer obtaining content from the distributor (an e-book store, a conditional access video store, etc). The consumer would then render payment to the clearing-house for a license to use the content. This license, combined with any necessary decryption keys, would then be bound to the content making it usable in the ways described by the rights package. Clearly, the role of the distributor and clearing-house is to enable an electronic business model and complements the provider and consumer in a generic DRM system.

A closer inspection of the transactions involved in this model reveals the following:

i.) There is a requirement for a "rights expression language" (REL) that is comprehensive enough to handle a wide range of licensing possibilities yet scalable in the sense that a subset can be implemented on memory and processor limited devices.

ii.) There is a *critical* requirement for a trusted client platform. What happens to multimedia content once it is downloaded by a client for consumption? Can the "rights" associated with the content be stripped away? If so, the DRM fails completely. Hence, it is generally acknowledged [62][39][28] that there is a requirement to *trust* that the client can enforce the rights associated with the content and that it is extremely difficult for determined adversaries to free the content from the adversaries: "…only when content owners can trust that their assets are handled appropriately… will a DRM system be able to become successful" ([62] p. 66).

iii.) There are privacy concerns. A rights expression language is used to "express how (and by whom) digital items may be used" ([28] p. 86) and this implies the divulging of personal identity. The integration of a personal identity with digital media and the possible exploitation of this

information for business and marketing intelligence is generally viewed dimly by the public at large. And, as we saw earlier, it may be in the interest of a provider to bind consumer information to content that has been purchased in order to "fingerprint" the content and identify potential pirates. However, "users have a right (and a strong demand) not to lay traces in the network" ([62] p. 96). To assuage public mistrust, research into privacy concerns continues to be an ongoing field of research in relation to DRM (see [65],[64], [62], for example).

iv.) There is a need for standardization. The DRM "system" should be standardized to avoid "stove-pipe" designs that limit interoperability (unless that is the business aim, of course). As indicated by Koenen et al., "the content industry needs to deploy legitimate content services that compete favourably (based on features, not on price) with unauthorized free services. A Simple and seamless user experience must be part of that goal, and DRM interoperability is necessary to achieve it" ([39] p. 884).

In the sections that follow, we begin with a short discussion of the issues surrounding trusted computing and why it is turning out to be such an important yet difficult component of DRM. We then discuss two approaches that have emerged towards the design of standardized DRM (top down and bottom up). We close by considering purely legal approaches to managing DRM.

5.3 Trusted Computing

The authors of [18] provide a very client-centric definition of a DRM system. In their work, the responsibility of a DRM system is to ensure that:

- The client cannot remove the encryption from the file and send it to a peer;
- The client cannot "clone" its DRM system to make it run on another host;
- The client obeys the rules set out in the DRM license; and
- The client cannot separate the rules from the payload.

It should be obvious that the client used by an end user to consume digital content is actually a policy enforcement agent and is therefore, in fact, the "lynchpin" in the DRM system. Software based clients are inherently weak since an adversary can spend as much time and resource as is desired (or necessary) to reverse-engineer the code and determine how to unlock digital content from any DRM oriented restrictions placed on its use. Essentially, we must "rely on the security of the platform to ensure that the content is used in accordance with its associated license" ([17] p. 3). Technical means (such as code obfuscation) and legal approaches (such as the End-User License Agreements and copyright legislation) have so far proven ineffective in deterring software code reverse engineering. Therefore, any DRM client based on software alone will almost certainly be broken and have the content associated with it leaked into the internet.

This is not to say that there are no efforts toward implementing tamper-proof client devices. As described by Rob Enderl in [41], the Trusted Computing Group (TCG), a consortium of corporations both large and small, was formed specifically to develop open standards for hardware enabled trusted computing. However, he is quick to dispel the myth that the work of the TCG can be used for "spying or DRM" since the scope of the work does not currently encompass a requirements for DRM. However, the door is left open to a later application of this technology to DRM ends.

Indeed, Microsoft is currently engaged in designing what they call the "Next-Generation Secure Computing Base" (NGSCB) which was formerly known as Palladium [43]. The core DRM functionality is described in their "Digital Rights Management Operating System" patent application on January 1999: "To protect the rights-managed data resident in memory, the digital rights management operating system refuses to load an un-trusted program into memory while the trusted application is executing…" [42]. However, Microsoft has received a fair amount of negative publicity concerning this effort since it is viewed by many in the community as an effort to control hardware and software development and distribution [66].

While it is an important yet contentious component, a trusted client is only one aspect of a larger group of elements that must work together to provide a DRM infrastructure. We next consider two approaches to implementing these other elements.

5.4 Top Down: MPEG-21

The top-down approach to DMR is characterized by an effort to implement an all-encompassing standard that can be applied from the most complex devices to the most simple. This top-down approach is being championed by MPEG-21. At the time of writing, MPEG-21 is an incomplete standard DRM standard.

The Motion Picture Expert Group (MPEG) began seriously considering Intellectual Property Management and Protection (IPMP) in MPEG-4 (although IPMP appeared as early as MPEG-2). IPMP is MPEG's term for DRM [51]. The original MPEG-4 specification provided IPMP "hooks" into which proprietary DRM systems could be connected. [62] Among other things, these hooks provided mechanisms to associate an audio-visual object (AVO) to the unique ID of a proprietary IPMP system. An MPEG-4 compliant decoder could "download" any necessary IPMP plug-in needed to handle an AVO if it were not already available.

The MPEG-21 approach is much broader in scope than that of MPEG-4. The goal of MPEG-21 IPMP is ultimately to allow users to attach rights and conditions to Digital Items and to expect those rights to remain attached to those Digital Items across a wide variety of networks and end user devices [67]. Essentially, MPEG-21 is attempting to build the "big picture" of digital rights management. MPEG-21 seeks to understand, integrate, and standardize all of the disparate elements that exist now for DRM, to perform a gap analysis, and to fill in where standards appear to be lacking [68][51].

There are seven major parts to the MPEG-21 standard. These are summarized below from [68] and [69]:

1. Vision, technologies, and strategies.
2. Digital Item Declaration (DID).
3. Digital Item Identification (DII).
4. Intellectual Property Management and Protection (IPMP).
5. Rights Expression Language (REL).
6. Rights Data Dictionary (RDD).
7. Digital Item Adaptation (DIA)

The DID provides abstract ways of defining a digital item. To be successful, it must be capable of describing all current and future types of content. The DII provides a means of identifying a digital item (in whole or in part) in an unambiguous way. The DID and DII are currently international standards.

IPMP defines a framework for providing hooks to DRM tools and is an extension of the work started in MPEG-4. The DIA targets the provision of tools to adapt digital items in the interest of providing a user with "the best content experience for the content required" ([68 p. 67). Digital adaptation may be necessary to mask implementation details from end users.

The RDD and REL are two very closely related parts of the standard. The RDD is, quite simply, "a set of clear, consistent, structured, integrated, and uniquely identified terms to support the MPEG-21 REL" ([69] p. 57). The RDD, although it is a separate part, is really only a subcomponent of the REL.

The REL is quite possibly the most interesting aspect of the MPEG-21 standard and we will now examine it more closely. Later, we will present a competing REL developed through the bottom-up approach.

5.4.1 MPEG-21 Rights Expression Language

The REL provides a machine-readable language for declaring usage rights using the constructs provided in the RDD [68]. In response to a Call for Proposal by MPEG in 2001, ContentGuard [70] submitted their Extended Rights Markup Language (XrML) as a contender to form the core technology of the MPEG-21 REL standard. ContentGuard's XrML 2.0 has since been formally accepted as the basis of the REL which attained international standard status in 2004 [71].

The most important element in XrML based REL is the *license*. We briefly discussed the issuance of licenses to individuals during the description of a typical DRM model. Here, we examine the elements of this license in terms of the XrML based REL.

A *license* is made up of a number of components as shown in the figure below (Figure 3 of [72]).

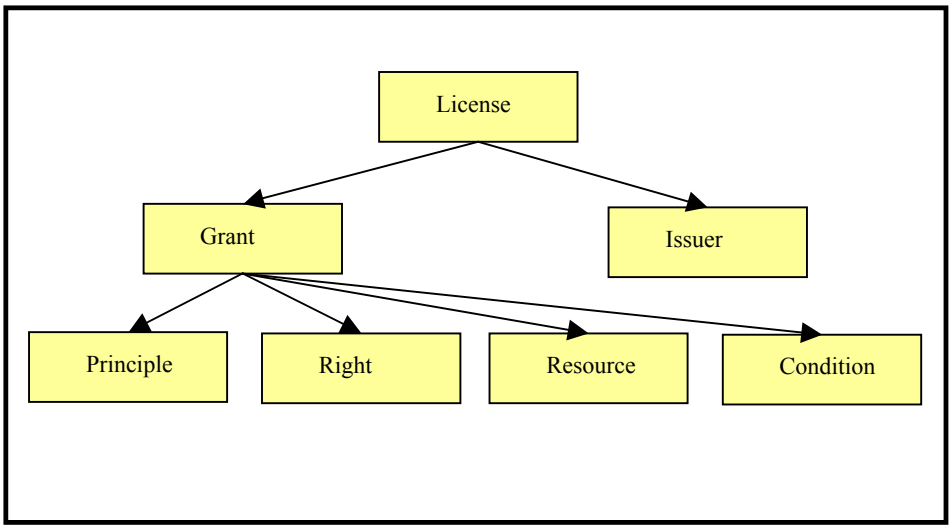

Figure 13: MPEG REL Data Model (Figure 3 of [72])

A license will identify an *issuer* and a *grant*. The issuer identity is specified by the use of public key certificates and signatures. A license will specify at least one grant that "conveys to an identified party the right to use a resource subject to certain conditions" ([73] p. 1). In turn, the grant identifies a *principal* to whom the grant applies, a *resource* against which the grant is to be applied, the *right* that the grant confers, and any applicable *conditions* that may apply.

Each principle is identified by some authentication mechanism. Typically, this could be performed using public key authentication mechanisms that can leverage the large established base of technology and knowledge. It is interesting to note here that a full public key "infrastructure" is not necessarily implied by the use of public key cryptography. For instance, when purchasing content for the first time from an e-retailer, the transaction might require the principal (or consumer) to provide his/her public key as part of the transaction. In the same manner as PGP, this public key would be verified out-of-band and then could be retained by the retailer against the customer's identification for future reference. In this situation, there is no "infrastructure" to worry about since the retailer is not providing the customer with a digital identity.

Each right conferred by a grant "specifies an action or activity that a principal may perform using a resource" ([73] p. 2). Examples include the right to play a movie, print a document, copy an MP3, etc.

Finally, conditions are attached to grants to limit the scope of some or all of the rights contained in the grant. Examples include the right to play a movie *only five times* or *only between two dates*. Conditions may also specify pre-requisites that a principal must possess before a right can take effect.

5.5 Bottom Up: Mobile DRM

In this section, we consider the class of proprietary DRM solutions. More specifically, we will examine the approach taken by the Open Mobile Alliance Mobile DRM (OMA MDRM).

As noted in [28], there are a number of "domains" in which proprietary DRM solutions (or elements of DRM solutions) have been developed. These include the broadband (mostly music services) and broadcast (mostly conditional access systems) domains. In each domain, the DRM solutions are generally proprietary and are tailored to specific technologies or applications. For example, many music services (for example, BuyNow, Musicmatch, etc) are built upon Microsoft's Windows Media Audio format while others (for example, Apple iTunes) use an MPEG codec combined with a proprietary DRM technology [39].

In the mobile domain, the OMA, which was founded in 2002, set as its goal to "introduce open standards and specifications based upon market and customer requirements for mobile industry" ([74] p. 3). In this sense, it is a mobile parallel to the MPEG-21 effort. However, while MPEG-21 attempts to provide an all-encompassing standard, OMA has scoped their business model and standard to meet the capabilities of a limited range of terminal device. This approach has lead to very fast turnaround from initial design to final deployment of the first version of the standard. OMA MDRM 1.0 has enjoyed a very rapid market adoption rate due to the very fact that it targets a minimal terminal device (typically cell phones) along with a very limited number of content distribution models [75].

The OMA MDRM 2.0 specification, released in early 2004, applies to more complex devices and adds a number of security features (for example, public key encryption for integrity and confidentiality) and expanded business models [75]. As the standard continues to grow to address ever more powerful mobile devices, it may eventually "jump ship" to non-mobile devices. It is not inconceivable that, in the not too distant future, it will either compete directly with or must become compatible with MPEG-21.

5.6 Legal Approaches

This is the final section in this review of digital rights management. While it is not a technical approach or an attempt to put forth a standard, it does represent a significant element in DRM. There have been a number of legal approaches for the enforcement of digital rights management [40]. For the most part, these approaches make it an illegal act to reverse engineer the copy protection or DRM schemes applied to digital content.

The prevailing legal approach (in the United States of America) has been the Digital Millennium Copyright Act (DMCA). The DMCA is an American copyright law that was passed in 1998 in order to implement obligations under the World Intellectual Property Organization (WIPO) [76][40]. However section 1201 of the DMCA specifically

addressed concerns by copyright owners about the piracy of their work. This section specifically bans acts that circumvent access control mechanisms and the distribution of tools or technologies that may be used for such circumvention [76]. However, there have been a large number of un-intentional side effects of the DMCA in which the law has been used in ways not envisioned by those who drafted it. A summary of a few of these follows (all of the examples below are from [76]):

- The DMCA has been used in a number of cases to stifle academic research. For example, in 2000, the Secure Digital Music Initiative (SDMI) issued a public challenge to defeat music watermarking technologies. Professor Edward Felton and researchers at Princeton, Rice, and Xerox took up the challenge and were successful in defeating the watermarks (by removal). However, when they went to present their work, they were blocked by SDMI representatives through a threat letter invoking the DMCA. As a result of this, [76] asserts that a number of other research scientists and organizations have either curtailed or completely abandoned research that might come under attack by the DMCA. Even the venerable IEEE was spooked into implementing a policy in 2001 that would force researches to indemnify them from liability under the DMCA (the policy was eventually revoked).

- The DMCA has been used to enforce sensorship on the world wide web. 2600 Magazine was forced, using provisions of the DMCA to remove all reference to DeCSS code from their web site. DeCSS was used to circumvent CSS copy protection for DVDs ultimately so that the DVDs could be played by "non-DRM compliant" players on the Linux operating system.

- Finally, the DMCA has been used as a weapon of good business acumen. In 2003, Lexmark invoked the DMCA to prevent the chip manufacturer Static Control from selling chips that were needed by after-market toner cartridge makers. After-market toner cartridge vendors were producing and selling toner cartridges for Lexmark printers that Lexmark could not match. Lexmark added some trivial "authentication routines" to the chips inside each cartridge and when Static Control reverse engineered these chips, Lexmark used the circumvention clauses of the DMCA to file an injunction preventing further manufacture.

Other legal approaches listed by [40] include the Security Systems Standards and Certification Act (SSSCA), the European Union Copyright Directive (EUCD) and the Copyright Amendment Act (Australia). The SSSCA is a proposal started in 2002 to force hardware manufacturers in the U.S. and any importers to implement DRM technology into any device hosting a processor and capable of handling multimedia content. The SSSCA appears to still be in draft bill status.

6 Conclusion

In this paper, we reviewed a broad range of topics related to the security of multimedia content. Our investigation began with an analysis of security mechanisms for the protection of content being streamed in conditional access systems. These mechanisms included the use of encryption to provide confidentiality services. We also looked at some interesting approaches to providing effective and scalable key management since this is typically considered outside of the scope of most work encryption-based research.

We briefly reviewed watermarking as a technique to provide proof of ownership and as a mechanism for fingerprinting multimedia content (for tracking down those engaged in illegal distribution of content). We examined the properties that could be expected of various watermarks and then we studied a number of interesting attacks against watermarks which appear in the literature as driving forces for further research.

Finally, we took a very brief look at digital rights management as an essential and emerging element of multimedia content protection. This brief look covered promising DRM approaches such as OMA DRM and MPEG-21. We completed the review with a short foray into attempts to legislate DRM.

References

[1] B. Bharagava, C. Shi, and S. Wang, "MPEG video encryption algorithms," *Multimedia Tools and Applications*, vol. 24, pp. 57-79, Sept 2004.

[2] Y.T. Chan, Wavelet Basics, Kluwer Academic Publishers, 1995.

[3] A. Eskicioglu, J. Town, E. Delp, "Security of digital entertainment content from creation to consumption," *Signal Processing: Image Communication*, vol. 18, pp. 237-262, February 2003.

[4] A. Eskicioglu, "Multimedia security in group communications: recent progress in wired and wireless networks," *Proceedings of the IASTED International Conference on Communications and Computer Networks*, pp. 125-133, Cambridge, MA., November 4-6, 2002.

[5] A. Eskicioglu, S. Dexter, and E. Delp, "Protection of multicast scalable video by secret sharing: simulation results," *Proceedings of SPIE Security and Watermarking of Multimedia Content V*, vol. 5020, pp. 505-515, Santa Clara, CA, January 21-24, 2003.

[6] A. Eskicioglu, "Multimedia security in group communications: recent progress in key management, authentication, and watermarking," *Multimedia Systems*, vol. 9, pp. 239-248, 2003.

[7] A. Eskicioglu and E. Delp, "A key transport protocol based on secret sharing applications to information security," *IEEE Transactions on Consumer Electronics*, vol. 48, no. 4, pp. 816-824, November 2002.

[8] A. Eskicioglu and M. Eskicioglu, "Multicast security using key graphs and secret sharing," *Proceedings of the Joint International Conference on Wireless LANs and Home Networks (ICWLHN 2002) and Networking (ICN 2002)*, pp. 228-241, Atlanta, GA, August 26-29, 2002.

[9] YL Huang, S. Shieh, FS Ho, and JC Wang, "Efficient key distribution schemes for secure media delivery in Pay-TV systems," IEEE Transactions on Multimedia, vol. 6, no. 5, October 2004.

[10] H. Radha, M. van der Schaar, and Y. Chen, "The MPEG-4 fine-grained scalable video coding method for multimedia streaming over IP," *IEEE Transaction on Multimedia*, vol. 3, no. 1, March 2001.

[11] S. Wee, and J. Apostolopoulos, "Secure scalable video streaming for wireless networks," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Salt Lake city, Utah, May 2001.

[12] S. Wee, and J. Apostolopoulos, "Secure scalable video streaming and secure transcoding with jpeg-2000," Hewlett Packard Labs (www.hpl.hp.com/techreports) Tech Report HPL-2003-110, 2003. Available at www.hpl.hp.com/techreports/2003/hpl-2003-117.html.

[13] W. Trappe, J. Song, R. Poovendran, and K. Liu, "Key management and distribution for secure multimedia multicast," *IEEE Transactions on Multimedia*, vol. 5, no. 4, December 2003.

[14] Deleted (duplicate).

[15] Deleted (duplicate)

[16] A. Vetro, C. Chritopoulos, H. Sun, "Video transcoding architectures and techniques: an overview," *IEEE Signal Processing Magazine*, pp. 18-29, March 2003.

[17] S. Haber, B. Horne, J. Pato, T Sander, R. Tarjan, "If Piracy is the Problem, Is DRM the Answer?", Hewlett Packard Labs (www.hpl.hp.com/techreports) Tech Report HPL-2003-110, 2003. Available at www.hpl.hp.com/techreports/2003/hpl-2003-110.html.

[18] P. Biddle, P. England, M. Peinado, and B. William, "The darknet and the future of content distribution," *2002 ACM Workshop on Digital Rights Management*, 2002. Available at www.cs.purdue.edu/homes/ayg/cs590a.

[19] X. Liu, A. Eskicioglu, "Selective encryption of multimedia content in distribution networks: challenges and new directions," *IASTED International Conference on Communications, Internet and Information Technology (CIIT 2003)*, Scottsdale, AZ, November 17-19, 2003.

[20] A. Eskicioglu, E. Delp, "An integrated approach to encrypting scalable video," *Proceedings of the IEEE International Conference on Multimedia and Expo*, pp. 573-576, Lausanne, Switzerland, August 26-29, 2002.

[21] Video Compression Tutorial, www.wave-report.com/tutorials/vc.htm (attached).

[22] E. Lin, G. Cook, P. Salama, and E. Delp, "An overview of security issues in streaming video," *Proceedings, International Conference on Information Technology: Coding and Computing*, pp. 345-348, April 2001.

[23] C. Venkatramani, P. Westernink, O. Versheure, and P. Frossard, "Securing media for adaptive streaming," *Proceedings of the Eleventh International Conference on Multimedia*, pp. 307-310, 2003.

[24] P. Tudor, "MPEG-2 video compression," *Electronics and Communication Engineering Journal*, pp. 257-264, December 1995.

[25] M. Marcellin and A. Bilgin, "JPEG2000: Highly scalable image compression," *Proceedings of 2001 International Conference on Information Technology: Coding and Computing (ITCC2001)*, pp. 268-272, Las Vegas, Nevada, 2001.

[26] Y. Huang, S. Shieh, F. Ho, and J. Wang, "Efficient key distribution schemes for secure media delivery in pay-TV systems," IEEE Transactions on Multimedia, vol. 6, no. 5, October 2004.

[27] C. Wong, M. Gouda, and S. Lam, "Secure group communications using key graphs," *IEEE/ACM Transactions on Networking*, vol. 8, no. 1, pp. 16-30, February 2000.

[28] W. Jonker and JP Linnartz, "Digital rights management in consumer electronics products," *IEEE Signal Processing Magazine*, pp. 82-91, March 2004

[29] M. Barni and F. Bartolini, "Data hiding for fighting piracy," *IEEE Signal Processing Magazine*, pp. 28-39, March 2004.

[30] D. Kundur and K. Karthik, "Video fingerprinting and encryption principles for digital rights management," *Proceedings of the IEEE*, vol. 92, no. 6, June 2004.

[31] A. Kejariwal, "Watermarking," *IEEE Potentials Magazine*, pp. 37-40, October/November 2003.

[32] F. Petitcolas, R.J. Anderson, and M. Kuhn, "Information hiding – a survey," *Proceedings of the IEEE*, pp. 1062-1078, vol. 87, no. 7, July 1999.

[33] F. Petitcolas, and R.J. Anderson, "Evaluation of copyright marking systems," *Proceedings of IEEE Multimedia Systems'99*, vol. 1, pp. 574-579, 7-11 June 1999.

[34] F. Hartung and M. Kutter, "Multimedia watermarking techniques," *Proceedings of the IEEE*, vol. 87, no. 7, July 1999.

[35] M. Barni ed. "What is the future for watermarking? (part II)," *IEEE Signal Processing Magazine*, pp. 53-57, November 2003.

[36] S. Craver, N. Memon, B.L. Yeo, and M. Yeung, "Resolving rightful ownerships with invisible watermarking techniques: limitations, attacks, and implications," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 4, May 1998.

[37] A. Adelsbach, S. Katzenbeisser, and H. Veith, "Watermarking schemes provably secure against copy and ambiguity attacks," *Proceedings of the 2003 ACM Workshop on Digital Rights Management*, pp. 111-119, October 2003.

[38] M. Ramkumar and A. Akansu, "A robust protocol for proving ownership of multimedia content," *IEEE Transactions on Multimedia*, vol. 6, no. 3, June 2004.

[39] R. Koenen, J. Lacy, M. Mackay, and S. Mitchel, "The long march to interoperable digital rights management," *Proceedings of the IEEE*, vol., 92, no. 6, June 2004.

[40] Q. Liu, R. Safavi-Naini, and N. Sheppard, "Digital rights management for content distribution," *Proceedings of the Australasian Information Security Workshop Conference on ACSW Frontiers 2003*, vol. 21, pp 49-58, 2003.

[41] R. Enderle, "Trusted computing: maligned by misrepresentations and creative fabrications," Storage Pipeline E-Magazine, 02 May 2004 (attached).

[42] T. O'Reilly, "Microsoft patents 'Digital Rights Management Operating System", 13 Dec, 2001, available at www.oreillynet.com/cs/user/view/wlg/956 (attached).

[43] K. Coyle, "Digital Rights Management – Part 4". Available at www.kcoyle.net/drm_basics4.html (attached).

[44] Deleted (duplicate)

[45] W. Geckle, "MPEG Video Compression, Lecture 10," available at www.apl.jhu.edu/Classes/Geckle/525759

[46] W. Geckle, "MPEG Video Compression, Lecture 9," available at www.apl.jhu.edu/Classes/Geckle/525759

[47] E. Raymond, ed., The New Hacker's Dictionary, 3rd Edition, October 1996, MIT Press.

[48] b-bstf, "A guide to internet piracy," 2600 Magazine, pp. Summer 2004.

[49] Xxxx Xxxxxxxx, "A day in the life of a warez broker," Phrack 47 (available at www.phrack.org).[50]"MPEG: achievements and current work," November 2001, available at www.chiariglione.org/mpeg/mpeg_general.htm.

[51] R. Koenen, "Object-based MPEG offers flexibility," *EE Times*, 12 November, 2001 (attached). Available at www.eetimes.com.

[52] Jia-Woei David Chen www.personal.psu.edu/users/j/u/juc169/fall2004 ee485chen_proj0.htm[53] A. Shamir, "How to share a secret," *Communications of the ACM*, vol.22, no. 11, pp. 612-613, November 1979.
[54] F. Duan, I. King, "A short summary of digital watermarking techniques for multimedia data," *Proceedings of the 1999 Hong Kong International Computer Conference, Hong Kong, 1999* (available at www.cs.cehk.edu.hk/~king/publications.htm)

[55] H. Jin, J.Lotspiech, S. Nusser, "Marking and tracing methods: Traitor tracing for prerecorded and recordable media," *Proceedings of the 4th ACM Workshop on Digital Rights Management*, pp 83-90, October 2004.
[56] J. Nagra, C. Thomborson, and C. Collberg, "A functional taxonomy for software watermarking," *Proceedings of the 25th Australasian Conference on Computer Science*, vol. 4, pp. 177-186, 2002.

[57] A. Kahng, J. Lach, W. Mangione-Smith, S. Mantik, I. Markov, M. Potkonjak, P. Tucker, H. Wang and G. Wolfe, "Watermarking techniques for intellectual property protection," *Proceedings of the Design Automation Conference '98*, pp. 776-781, 1998.

[58] S. Katzenbeisser and J. Dittmann, "Malicious attacks on media authentication schemes based on invertible watermarks." *Proceedings of the SPIE vol 5306, Security and Watermarking of Multimedia Contents VI*, 2004, pp. 838-847.

[59] Definition of Digital Rights Management, whatis.com (attached). Available at http://whatis.techtarget.com/definition/0,289893,sid9_gci493373,00.html.

[60] M. Lesk, "The good, the bad, and the ugly. What might change if we had good drm," *IEEE Security and Privacy*, pp. 63-66, May/June 2003.

[61] J. Camp, "Access denied," *IEEE Security and Privacy*, pp. 82-85, Sept/Oct 2003.

[62] N. Rump, "Can digital rights management be standardized? An overview of DRM standardization within MPEG," *IEEE Signal Processing Magazine*, pp. 63-70, March 2004.

[63] R. Grimm, and P. Aichroth, "Privacy protection for signed media files," *Proceedings of the 2004 Multimedia and Security Workshop on Multimedia Security*, pp. 93-99, 2004.

[64] C. Conrado, F. Kamperman, G. Jan Schrijen, W. Jonker, "Privacy in an identity-based DRM system," *Proceedings of the 14th International Workshop on Database and Expert Systems Applications*, pp. 389-395, Sept 2003.

[65] B. Park, J. Kim, and W. Lee, "PrecePt: a privacy-enhancing license management protocol for digital rights management," *Proceedings of the 18th International Conference on Advanced Information networking and Application*, vol. 1, pp. 574-579, March 2004.

[66] Definition, Next-Generation Secure Computing Base, from Wikipedia,The Free Encyclopedia (en.wikipedia.org/wiki/palladium_operating_system), (attached).

[67] R. Koenen, "Intellectual property management and protection in MPEG standards," Jan 2001 (available at www.chiariglione.org/mpeg/standards/ipmp/).

[68] I. Burnett, R. Van de Walle, K. Hill, J. Bormans, F. Pereira, "MPEG-21: goals and achievements," *IEEE Multimedia*, vol. 10, no. 4, pp 60-70, Oct-Dec 2003.

[69] J. Bormans, J. Gelissen, and A. Perkis, "MPEG-21: the 21st century multimedia framework," *IEEE Signal Processing Magazine*, March 2003, pp. 53-62.

[70] www.contentguard.com.

[71] DRMWatch Staff, "ISO Approves MPEG REL," April 2004, (attached) www.drmwatch.com/standards/article.php/3334611.

[72] X. Wang, "MPEG-21 rights expression language: enabling interoperable digital rights management," *IEEE Multimedia*, pp. 84-87, Oct-Dec 2004.

[73] ContentGuard basic feature examples: Grant (attached). Available at www.contentguard.com/MPEGREL_referencelisting.asp?type=basicfeatureexamples.

[74] Mobile Digital Rights Management White Paper, Sonera MediaLab, August 2003 (attached). Available at www.medialab.sonera.fi/workspace/moviledrmwhitepaper.pdf.

[75] B. Rosenblatt, "Open mobile alliance announces version 2.0 of DRM standard," Feb 2004, (attached). Available at www.drmwatch.com/standards/article.php/3308861.

[76] "Unintended Consequences: Five years under the DMCA" (attached). Available at www.eff.org/IP/DMCA/unintended_consequences.pdf.

Attachments

Tab	Attachment
1	Video Compression Tutorial, www.wave-report.com/tutorials/vc.htm
2	R. Enderle, "Trusted computing: maligned by misrepresentations and creative fabrications," Storage Pipeline E-Magazine, 02 May 2004
3	T. O'Reilly, "Microsoft patents 'Digital Rights Management Operating System", 13 Dec, 2001, available at www.oreillynet.com/cs/user/view/wlg/956
4	K. Coyle, "Digital Rights Management – Part 4". Available at www.kcoyle.net/drm_basics4.html
5	R. Koenen, "Object-based MPEG offers flexibility," *EE Times*, 12 November, 2001
6	Definition of Digital Rights Management, whatis.com. Available at http://whatis.techtarget.com/definition/0,289893,sid9_gci493373,00.html
7	Definition, Next-Generation Secure Computing Base, from Wikipedia,The Free Encyclopedia (en.wikipedia.org/wiki/palladium_operating_system).
8	DRMWatch Staff, "ISO Approves MPEG REL," April 2004
9	ContentGuard basic feature examples: Grant. Available at www.contentguard.com/MPEGREL_referencelisting.asp?type=basicfeatureexamples
10	Mobile Digital Rights Management White Paper, Sonera MediaLab, August 2003.
11	B. Rosenblatt, "Open mobile alliance announces version 2.0 of DRM standard," Feb 2004.
12	Unintended Consequences: Five years under the DMCA. Available at www.eff.org/IP/DMCA/unintended_consequences.pdf.