Video Fingerprinting and Encryption Principles for Digital Rights Management

DEEPA KUNDUR, SENIOR MEMBER, IEEE, AND KANNAN KARTHIK, STUDENT MEMBER, IEEE

Invited Paper

This paper provides a tutorial and survey of digital fingerprinting and video scrambling algorithms based on partial encryption. Necessary design tradeoffs for algorithm development are highlighted for multicast communication environments. We also propose a novel architecture for joint fingerprinting and decryption that holds promise for a better compromise between practicality and security for emerging digital rights management applications.

Keywords*—*Digital fingerprinting tutorial, digital video encryption survey, joint fingerprinting and decryption (JFD), video scrambling.

I. Introduction

This paper investigates multimedia security algorithms that enable digital rights management (DRM) in resource constrained communication applications. Our focus is on the video-on-demand (VoD) business model, in which subscribers to a content-providing service request and receive video information at scheduled intervals. We consider situations in which on the order of hundreds or even thousands of users may wish near-simultaneous access to the same video content. Thus, for superior scalability the network service provider must transmit the content by making use of a multicast distribution model.

We focus on the problems of video fingerprinting and encryption. Fingerprinting, which was first introduced by Wagner [1] in 1983, is the process of embedding a distinct set of marks into a given *host* signal to produce a set of fingerprinted signals that each "appear" identical for use, but have a slightly different bit representation from one another. These differences can ideally be exploited in order to keep track of a particular copy of the fingerprinted signal. The marks, also called the fingerprint *payload*, are usually embedded through the process of robust digital watermarking. In digital watermarking, subtle changes are imposed on the host signal such that the perceptual content of the host remains the same, but the resulting composite watermarked signal can be passed through a detection algorithm that reliably extracts the embedded payload. In contrast, video encryption has the goal of obscuring the perceptual quality of the host signal such that access to the content is denied. In comparison to traditional cryptographic algorithms, those for video may often be "lightweight" in order to accommodate computational complexity restrictions; the term "video scrambling" is often used to refer to such processes.

The main objective of fingerprinting and encryption in a DRM context is to protect video content from a set of attacks applied by one or more attackers. We define an attacker as any individual who attempts to use a given piece of content beyond the terms, if any, negotiated with the content provider. Common attacks on video data include illegal access and tampering. Our work focuses on the problem of piracy, the illegal duplication and redistribution of content; we call such an attacker a *pirate*.

Overall, the objectives of this paper are twofold:

1) to present a state-of-the-art review and tutorial of the emerging areas of video fingerprinting and encryption highlighting design challenges for multicast environments;
2) to propose the approach of *joint fingerprinting and decryption* (JFD) to establish a better compromise between practicality and security for DRM applications.

Section II introduces some general security architectures for video applications. Sections III and IV introduce and survey the areas of video fingerprinting and encryption, respectively, demonstrating the necessary compromises for algorithm and system design. Section V proposes a novel joint fingerprinting and encryption framework that overcomes many of the obstacles of previous architectures;

Manuscript received September 14, 2003; revised December 15, 2003.

D. Kundur is with the Department of Electrical Engineering, Texas A&M University, College Station, TX 77843-3128 USA (e-mail: deepa@ee.tamu.edu).

K. Karthik is with the Edward S. Rogers Sr. Dept. of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 3G4, Canada (e-mail: karthik@comm.utoronto.ca).

Digital Object Identifier 10.1109/JPROC.2004.827356

preliminary algorithmic ideas are discussed. General conclusions are presented in Section VI.

II. Security Architectures

We consider a single transmitter which may be a VoD server that we refer to as the *source* or *server* that communicates with $n > 1$ receivers that we call *users*. In all situations, the source is responsible for embedding the global (i.e., static for all users) group watermark W_s that may contain copyright and ownership information, and is also responsible for encrypting the media content using secret key cryptography with a *group key* K_g that is common for all users. The use of a single group key for encryption under certain conditions can enable multicast communications, but requires more sophisticated key management.

At the receivers, each user must decrypt the content individually. Fingerprinting can occur either at the transmitter or receiver, and separate or integrated with the cryptographic process which is the basis for our architecture classifications. Fingerprint detection is assumed to occur offline at a later time outside the scope of the multicast communication setup. The reader should note that the watermarking process for W_S (which may be optional) is distinct from fingerprinting.

A. Transmitter-Side Fingerprint Embedding

In this approach, introduced in [2], the fingerprint is embedded at the source. An optional copy control or ownership watermark W_s is first embedded into the host media. Then a distinct fingerprint is marked in each copy of the media to be delivered to each of the n customers. Every watermarked and fingerprinted copy X_i for $i = 1, 2, \ldots, n$ is then encrypted separately using the same group key K_g (that is known at the source and by all the users) to produce Y_i, $i = 1, 2, \ldots, n$. Fig. 1(a) summarizes the approach.

One characteristic of this method is that n different copies of the media have to be simultaneously transmitted, which represents bandwidth usage of order $O(n)$. In addition, the number of copies of the media that must be encrypted and fingerprinted also increases linearly with n. Thus, the overall method suffers from poor scalability and cannot exploit the multicast infrastructure. Many current methods for fingerprinting implicitly make use of this architecture [3]–[5].

B. Receiver-Side Fingerprint Embedding

The next architecture, initially introduced in [6] with respect to digital TV and more recently discussed in [2] and [7] for DRM in digital cinema, involves fingerprinting at the receiver. In this scheme, shown in Fig. 1(b), the optional copyright watermark is embedded and the subsequent media is encrypted with the group key K_g to produce the encrypted content Y. Only one encryption (and no fingerprinting) is necessary at the server, reducing latency and complexity from the previous architecture. In addition, because only one signal needs to be transmitted to multiple users, multicast communications can be exploited.

At the receiver, the encrypted signal $\hat{Y}$ is decrypted by each user using K_g and is immediately fingerprinted with a mark f_i that is distinct for each user i to produce the fingerprinted media $\hat{X}_i$. For security, both decryption and fingerprinting must be implemented on a single chip or application-specified integrated circuit (ASIC) so that the decrypted signal is not easily accessible before fingerprinting. Furthermore, tamperproof hardware, which is difficult to achieve and still an open research problem, must be used in order to protect the purely decrypted host signal from eavesdropping.

The additional burden of fingerprinting at the receiver may be problematic if the transmission is real time. Consumers are not willing to pay excessively for security features that do not directly benefit them. Therefore, either low-complexity algorithms or nonreal-time implementations of fingerprint embedding are necessary, which may limit the use of this architecture for some applications.

C. Joint Fingerprinting and Decryption

In order to overcome the complexity issues of fingerprinting at the receiver while preserving the bandwidth, complexity, and latency efficiencies at the source, we propose the notion of *integrating* the decryption and fingerprinting processes. As discussed in the previous section, the server encrypts the media using the group key K_g. However, at each receiver a single secret key K_i that is unique for each user is employed for JFD. The process, in part, mimics decryption. However, the use of $K_i \neq K_g$ for decryption allows the introduction of a unique fingerprint for each user, making each decrypted copy distinct. The information carried by the fingerprint can be represented as the relative entropy between the source and decryption keys, that is, $H(f_i) = H(K_g|K_i)$, where $H(f_i)$ is the entropy of the fingerprint for user i, and $H(K_g|K_i)$ is the conditional entropy of the group key given the receiver's key. We present this approach in Fig. 1(c) and discuss a preliminary implementation in Section V. The structure of the scheme does away with the need for tamperproof hardware, but raises issues with respect to the tradeoff between imperceptibility and robustness of the fingerprint, especially with respect to collusion attacks.

D. Other Scalable Fingerprinting Architectures

Other methods have been proposed to overcome, in part, the scalability challenges of fingerprinting by *distributing* the fingerprinting process over a set of intermediate nodes such as routers. This shift in trust from the network source and destination extreme points to intermediate nodes creates a different set of challenges such as vulnerability to intermediate node compromise and susceptibility to standard network congestion and packet dropping. A more detailed discussion is beyond the scope of this paper; the reader is referred to [8] for a comparative survey by Luh and Kundur.

In the next section, we focus on the fingerprinting problem and introduce the coding and signal processing challenges of data embedding.

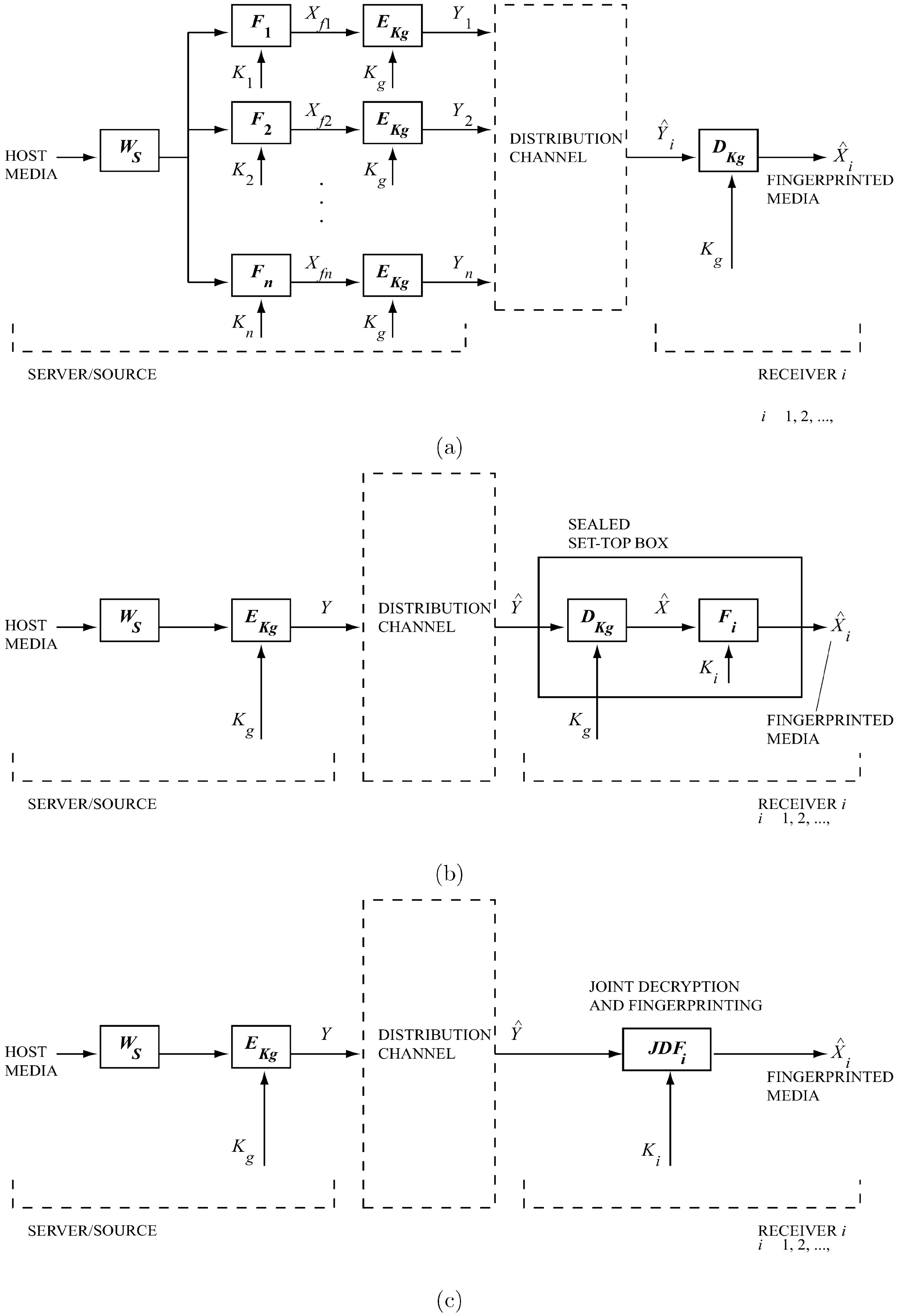

Fig. 1. Security models. (a) Transmitter-side fingerprint embedding. (b) Receiver-side fingerprint embedding (decryption and fingerprinting are decoupled). (c) Proposed JFD.

III. VIDEO FINGERPRINTING

A. General Formulation

We consider a video server that distributes fingerprinted copies of media to users $u_i \in U$ for $i = 1, 2, \ldots, n$ where u_i represents the ith user and U is the set of all users at a specific time in the media distribution system. A fingerprint f_i associated with user u_i is a binary sequence of length p. The set of all fingerprints associated for the users in U is denoted $F = \{f_1, f_2, \ldots, f_n\}$ with cardinality n

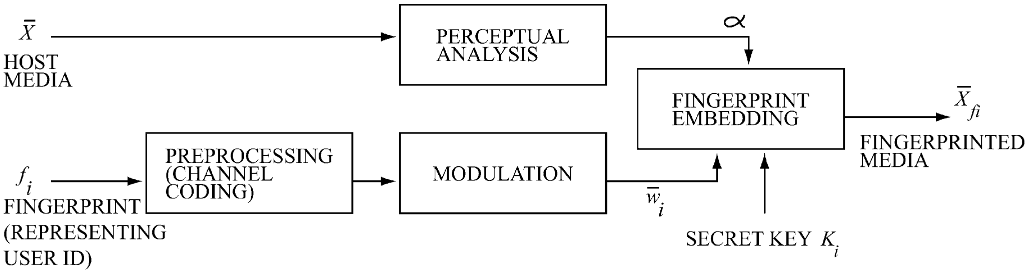

Fig. 2. Fingerprint embedding. The binary fingerprint is channel coded for robustness and then modulated so that it can be embedded in the host media. Perceptual analysis is conducted on the host in order to determine the depth or strength of embedding that provides a good tradeoff between imperceptibility and robustness.

and codeword length p bits. The fingerprint codewords are *modulated*, which involves the use of signal processing strategies to form a signal $\bar{w}_i \in W, i = 1, 2, \ldots, n$ that can be added imperceptibly to the host media with the help of human perception models. Fig. 2 shows the process of fingerprint embedding. The modulated version of the fingerprint is often called a watermark in the research literature, which has a distinct objective from the copyright watermark W_S previously discussed. For simplicity from now on our use of the term "watermark" unless otherwise specified will apply to the modulated fingerprint and not W_S.

In this context, a pirate is a user $u_p \in U$ who illegally redistributes his/her copy of the distributed media either in modified or unmodified form; an illegally distributed media is termed a *pirated copy*. In order to make sure that a pirated copy cannot be traced back to the pirate u_p, he/she will try to cover any tracks by attempting to erase the associated fingerprint f_p or frame another user. It is also possible that a subset of pirates $P \subset U$ may combine different copies of their fingerprinted media to achieve their goal. This powerful attack is called *collusion*. Collusion is also possible when a single user requests different copies from the server under different aliases (but this does not change the formulation of the problem). Fig. 3 summarizes where the general collusion attacks takes place in an overall media distribution system.

The server is responsible for secure video distribution, which is achieved through a combination of compression and security processing that includes fingerprinting. The server needs to balance efficiency of transmission and robustness of security without hindering the viewing experience of the users.

Once the server transmits the secured media over the distribution channel, it is received by all users consisting of two disjoint groups: the lawful users and the pirates. Lawful users consume the media in the manner that was agreed upon. Pirates tamper with the decrypted media, potentially collude in order to remove any fingerprints or frame other user(s), and insert the pirated copy into illegal distribution channels. If, at a later time, such a pirated copy is discovered, then it is sent to the source (or a party working with the source) and the associated fingerprint(s) are detected in order to trace the pirate(s). A fingerprinting scheme that consists of fingerprint generation, embedding, and detection stages must, therefore, address several challenges that we discuss in the next sections.

B. Fingerprinting Requirements

1) Fundamental Compromises: There is a basic tradeoff between fingerprint embedding and source coding. Compression attempts to remove redundancy and irrelevancy for the purpose of reducing storage requirements while maintaining the perceptual fidelity of the media. In contrast, fingerprinting shapes the irrelevancy within the media signal to transparently communicate security codes f_i along with the media. Thus, as first observed by Anderson and Petitcolas, if perfect compression existed, it would annihilate the process of fingerprinting [9]. The restricted structure of compression algorithms and limited accuracy of perceptual coding models, however, allows some irrelevant signal bandwidth to be used for fingerprinting. In general, the lower the required bit rate after compression, the smaller the length of the fingerprint that can be robustly embedded.

Another set of compromises exists between fingerprint capacity, defined as the total number of unique fingerprints that can be embedded and successfully distinguished at the receiver, and robustness, which reflects the inability for one or more pirates to erase or forge the fingerprint without affecting the commercial quality of the video. In general, the smaller the required fingerprint capacity, the easier it is through the effective use of redundancy to make the embedding robust. In general, the fingerprint capacity must ensure that all possible users can be serviced in the life cycle of the distribution system.

Similarly, there is a tradeoff between perceptual quality and robustness (or capacity). Transparency requires that the fingerprint be embedded in either perceptually irrelevant components such as high spatial frequencies or perceptually significant components with severely limiting amplitude. This is contradictory to the goals of robustness that aim to ideally embed high-energy fingerprint watermarks in perceptually significant components so that they cannot be easily removed [10].

Thus, one arrives at a set of compromises that must be resolved to design an effective media fingerprinting scheme. To summarize, these include the ability to work with existing

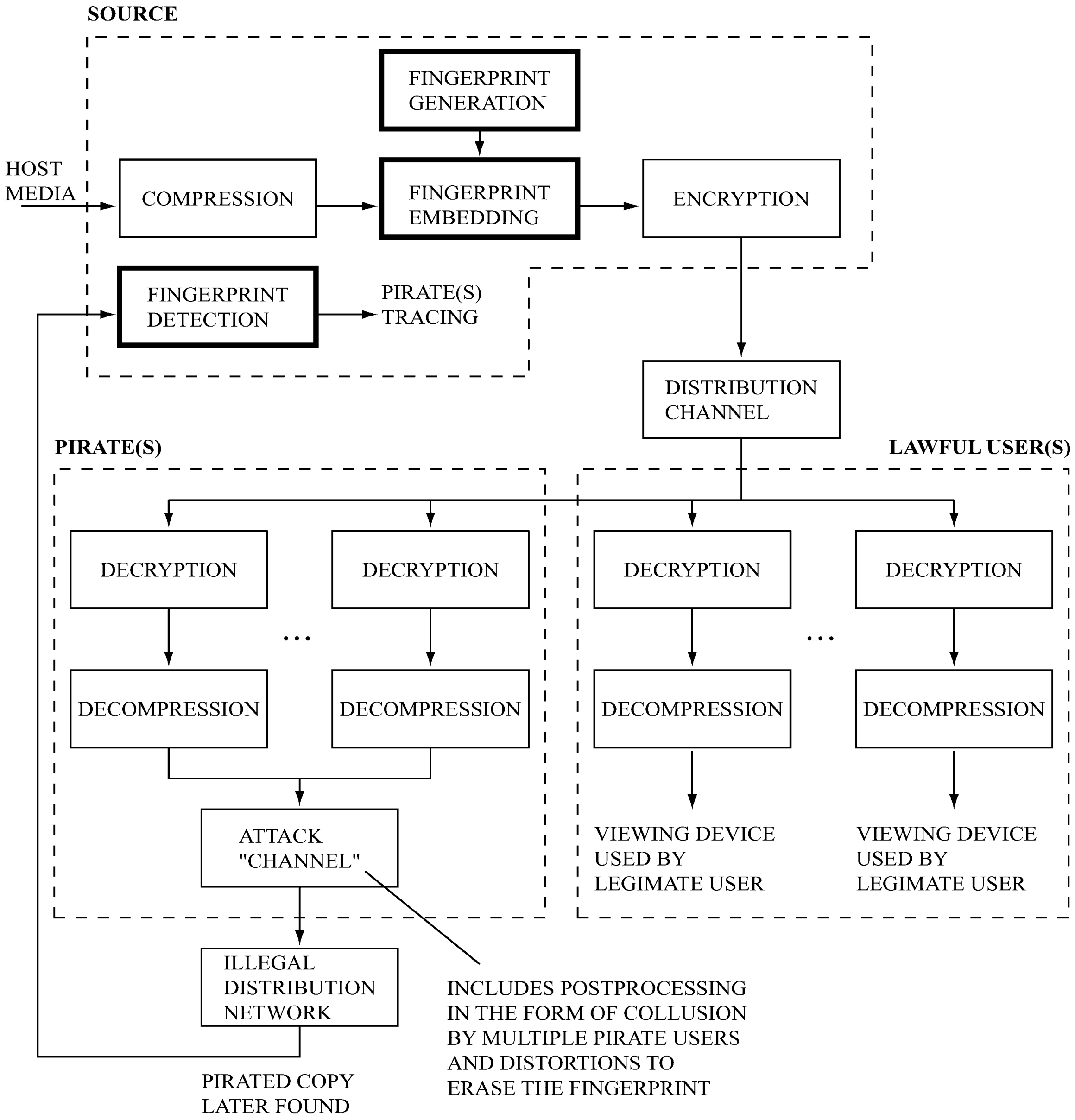

Fig. 3. Overview of video security, distribution, and piracy process. Fingerprinting consists of three stages: fingerprint generation, fingerprint embedding, and fingerprint detection as highlighted with bold boxes. We do not include copyright watermarking in this figure, which may be added prior to encryption, for reasons of simplicity. The compression process is placed prior to fingerprinting for practicality, although it may occur at other stages.

compression standards, robustness to signal processing and collusion-based attacks, capacity to handle unique detection of all possible users in the life cycle of the VoD system, and perceptual quality.

2) Practical Constraints: Other pragmatic issues involve restrictions on complexity, latency, quality of service (QoS) and bandwidth. For example, the server may receive hundreds of video requests within a short span of time, requiring real-time algorithm complexity. Related to this concept is hardware complexity that refers to computational resources, power, and memory utilization that must be minimal to reduce implementation costs. Bandwidth consumption is an additional concern for the content distributor, as Internet service provider (ISP) utilization costs may be nonnegligible. The final consideration must be that the superposition of the DRM architecture must not affect the end-to-end QoS of the distribution network.

From the perspective of the ISP, bandwidth efficiency is of primary concern. Therefore, fingerprinting schemes in which the associated watermark is embedded at the source, as discussed in Section II-A, do not scale well as the number of users grow. In contrast, more efficient architectures, presented in Sections II-B and II-C, are more suitable. Another issue of importance to the ISP is the level of DRM processing required from intermediate nodes or routers in the network. Ideally, an ISP would prefer to avoid the need for audit trails by using intermediate network nodes as discussed in Section II-D. Such strategies for DRM are not "network friendly," since there are issues related to trust and computational complexity at the inner nodes that increase network latency.

The end user, who has the most influence in determining the acceptance of a given media distribution and rights management system, requires a desired QoS for sufficiently limited power constraints. The fingerprinting cannot cause perceptual discomfort, for example, through watermark error propagation as discussed in [2]. Furthermore, in receiver-oriented fingerprinting schemes, the power consumed for fingerprinting must be minimal, especially in wireless environments.

C. Existing Work

The body of literature concerned with digital fingerprinting for DRM applications can be classified into three basic categories: coding theoretic, protocol enhancing, and algorithm specific. In coding-theoretic work, the authors view fingerprinting as a code design problem in which the codebook often represents the unique set of fingerprints (or a closely related quantity) that can be used for embedding. These formulations allow a well-structured modeling of the problem in order to isolate issues such as the necessary length of the fingerprint code, the number of users that may be supported by such a system, and the ability for users to collude to erase or fabricate a given fingerprint code. However, they make somewhat restrictive assumptions on the fingerprinting process and attackers' behavior. Early theoretical work by Blakely *et al.* [11] characterizes the constraints on a pirate with access to several distinct fingerprinted copies of data when attempting to erase or fabricate fingerprints. Boneh and Shaw [3] investigate the problem of fingerprint code design when one would like to restrict a maximum coalition of users, each carrying a distinct fingerprinted copy of the content, from colluding to fabricate another fingerprint (and, hence, frame another user for an unwanted act).

Protocol-enhancing methods provide protection against attacks on innocent users by the fingerprinting source. Pfitzmann and Schunter [12] introduce the notion of *asymmetric fingerprinting*, in which they investigate protocols to protect innocent users from being framed for piracy by the source. Pfitzmann and Waidner [13] address privacy issues by proposing a protocol that allows users to maintain their anonymity during content purchase, although they can be later identified if they distribute content illegally. The final class of algorithm-specific methods is the focus of our proposed work and is discussed in Section III-C3.

1) Fingerprint Generation, Embedding, and Detection: The fingerprinting problem can be broken down into several stages including fingerprint generation, embedding, and detection as highlighted in Fig. 3. Each of these components must be designed in order to keep the overall system robust to a given set of attacks. Fingerprint generation involves the design of a fingerprint "code" for every user that makes it possible to uniquely identify the code and, hence, the user during detection. Previous literature on code constructions [1], [3], [11] is useful for this stage in order to keep the fingerprints collusion-resistant under a set of conditions.

In addition, error correction code (ECC) strategies may be used to make the embedded codes more reliable in the face of attacks. The use of ECCs in the area of watermarking was motivated by the communication analogy of watermarking in which the processes of fingerprint embedding and detection are likened to modulation and signal reception in a communication system. Any attacks on the fingerprint characterize nonideal communication channel model often called the *watermark channel*. It then follows that for improved performance, fingerprint generation may use elements of channel coding to reduce the probability of detection error by creating interdependencies within the embedded codes at the price of increased complexity and bandwidth. Given the success of ECCs in achieving near-capacity information transmission, it has been concurred that such an approach is effective in improving the detection of fingerprints, although the tradeoffs for practical watermarking applications remain, in part, unexplored. Work by Baudry *et al.* [14] investigates the use of repetition and Bose–Chaudhuri–Hocquenghem codes (BCH codes) in improving the fingerprint robustness. The degree of tradeoff between robustness and code length is characterized assuming the watermark channel is an additive white Gaussian noise (AWGN) channel. Fernando *et al.* [15] investigate the effects employing channel codes such as block, convolutional, and orthogonal codes prior to watermark embedding. Their analysis shows the superior performance of convolution codes to the other coding approaches. One important question that still remains is how a designer can match a coding scheme with an embedding process to bolster performance over a broad class of attacks.

Fingerprint embedding employs signal processing methods to insert the fingerprints into the host signal such that the fingerprint is imperceptible yet robust to attacks including collusion. Detection complements this and takes into account the watermark channel characteristics to minimize watermark detection error. Several archetypes have been proposed for embedding and detection. The two main paradigms are based on spread spectrum signaling and quantization. Quantization methods [16], [17] are characterized by their ability to avoid host signal interference. That is, watermark detection under ideal conditions is guaranteed. Spread spectrum methods [18] are popular for their higher resistance to attacks modeled as narrow-band interference. However, given the vulnerability of spread spectrum approaches to fading-type attacks, other approaches such as communication diversity may be additionally adopted to supplement performance [17]. As previously discussed, all methodologies exhibit a compromise amongst fingerprint perceptibility, robustness, and capacity.

To be effective, the selection of the generation, embedding, and detection stages must match the type attack that will be applied to the marked media. We focus on the collusion attack, which is unique to fingerprinting applications; other watermarking applications in which only the static watermark W_S is embedded in the host for all users are not susceptible to collusion.

2) Collusion: Many popular signal processing attacks on watermarks have been modeled as random noise [18] or as fading [17].[1] These operations on the watermarked signal attempt to remove the watermark while maintaining perceptual fidelity. The location or some other random characteristic of the watermark, often called the *watermark key*, is used for watermark generation and embedding and is unknown to the attacker. This strategy, although not equivalent in security to cryptography, can provide basic protection for the embedded data. Security, however, may be significantly compromised if two or more copies of distinctly watermarked signals are available to the attacker. In such a case, it is possible that some portions of the watermark key, previously assumed to be unknown to the attacker, can be easily estimated. As the number of distinctly watermarked copies increases, it may be possible that a growing degree of information about the watermark key becomes known until the overall system is trivially broken.

Such an attack is called collusion. In collusion attacks, a group of users compare their distinctly fingerprinted copies to form another composite signal that either contains no fingerprint or frames an innocent user. We first focus on a popular subclass of collusion attacks, called *linear collusion* that works as follows:

$$\hat{X}_c = \sum_{i=1}^{t} \lambda_i \hat{X}_i \tag{1}$$

where $\hat{X}_c$ is the colluded (or composite) copy $\sum_{i=1}^{t} \lambda_i = 1$ and $\hat{X}_i$ is the fingerprinted copy of the video received by the ith user. It is possible that the originally fingerprinted copy at the source of user i (which we denote X_i) has undergone some small incidental distortions to produce $\hat{X}_i$ which is not exactly the same as X_i.

In (1) it is clear that X_c has elements of all fingerprinted copies of the colluders. It is possible, depending on the fingerprint generation, embedding, and detection stages, that one or more of the colluders can be identified from X_c. Intuitively, the fingerprinted copy of the user corresponding to the largest value of λ_i has the most influence on X_c and, hence, may be more easily identified from X_c. In many formulations of collusion, $\lambda_i = 1/t$ is employed as a condition of fairness among the colluders to equalize the probability that any of them are caught. Research on linear collusion for spread spectrum watermarking approaches of fingerprinting has been conducted and demonstrates how this attack is powerful when the correlation between watermarks is small and the number of colluders t is large [18].

General collusion attacks can be more deceptive and incorporate the various fingerprinted video $\hat{X}_i$ in a nonlinear fashion by, for example, taking the minimum, maximum, or median value of pixels in a given video stream location [19]. The only constraints on the pirate is that the X_c be perceptually identical to the family of fingerprinted signals X_i, and that the attack be computationally feasible. This means that the cost of the attack to the pirate is lower than the value of the video content.

3) Collusion-Resistant Fingerprinting Techniques: In this section, we discuss three contributions that fall under the class of algorithm-specific fingerprinting schemes. One of the first papers addressing collusion-resistant fingerprint design was by Dittmann *et al.* [20]. Their objective is to trace the exact subset of colluders from a total of n users when the number of colluders is equal to or below a prespecified threshold number t. In the same spirit as [1], [3], and [11], the authors assume that collusion occurs when the colluders compare distinctly fingerprinted signals, identify all the locations that differ for at least one pair of colluders, and then modify or remove the fingerprint from those positions of the media. Given this model, the task of collusion-resistant design becomes one of embedding fingerprints such that there are common elements among any t or fewer subsets of users so if any of these groups collude to form X_c, the common elements of their fingerprint (that still remain in the media by assumption) identifies them. Dittmann *et al.* demonstrated how judiciously developed watermarks could be employed to achieve this task. For instance, for $n = 3$ and $t = 2$, the goal is to identify one or two colluders from an overall set of three system users. This can be achieved by having a common watermark locations between each possible pair (since $t = 2$) of users (u_i, u_j) for $(i, j) = \{(1,2), (2,3), (1,3)\}$ and a unique location for each user i for $i = 1, 2, 3$. To design the collusion-resistant watermark and its locations for each user, the authors of [20] make use of the concept of finite geometries and projective spaces. For the case of $n = 3$ and $t = 2$, this involves finding three intersecting lines in a two-dimensional projective space (in this case, a plane). Every point in the projective spaces pseudorandomly maps to a location in the media. The intersecting points of any two lines in the projective space represents the common watermark locations of a pair of users that identifies the colluding pair. The method is extended for general values of t and n in which higher dimensional projective spaces are used. The main challenge with this approach is that a large number of fingerprint marking locations are necessary for large n. Thus, the scheme does not scale easily when involving short host video clips with fewer possible embedding locations.

Trappe *et al.* [5] investigate orthogonal and code division multiple access (CDMA)-type modulation scenarios for fingerprinting in the presence of collusion. In orthogonal modulation, each user is assigned a fingerprint that is orthogonal to all other fingerprints embedded in the signal; the fingerprint may be a pseudorandom noise (pn) sequence. Detection ideally requires the correlation of the received signal with all possible fingerprints where the highest correlation result identifies the embedded fingerprint. Thus, detection complexity scales linearly with the number of users in the system. Trappe *et al.* [5] show that the computational cost can be reduced at some expense of performance with coded modulation. Here, CDMA-type modulation is used in which every user has the same pn sequence that is modulated using a unique (for each user) binary code that are generated to be collusion-resistant using the theory of *balanced incomplete block designs* (BIBD) and the resulting fingerprints are termed anticollusion codes

[1] The reader should note that we are not including geometric or protocol attacks among others in our discussion.

(ACCs). Analysis and simulations verify the improved performance of the scheme to practical linear collusion attacks. The challenges involve overcoming the assumption that linear collusion is equivalent to a logical AND of the ACCs, which is not the case for more than two colluders, and the nontrivial nature of designing BIBD-codes for ACCs of arbitrary system parameters.

In contrast, Su *et al.* [21], [22] analyze the necessary conditions for the problem of resistance against linear collusion among frames within the same video sequence. The authors focus on two types of collusion: "Type I," in which the colluded result $\hat{X}_c$ is used to estimate the fingerprint in a given host video (and, hence, subtract it out), and "Type II," in which $\hat{X}_c$ is an estimate of the host where the fingerprint has been filtered out. The authors derive watermark design rules to ensure neither Type I and Type II collusion can be achieved by an attacker. The rules state that under a given set of conditions, the watermarks embedded in each frame must have proportional correlations to their host video frame counterparts. Thus, as the similarity amongst the host video frames changes temporally, the correlation of the associated watermarks must also follow a comparable evolution. To demonstrate the utility of the design rules, the authors develop the Spatially Localized Image-Dependent (SLIDE) video watermarking method [23]. A feature extraction algorithm is designed which selects *anchor points* around which components of the watermark are embedded. As the host video frame evolves in time, the locations of the anchor points also change such that similar frames contain similar watermarks and diverse frames contain diverse watermarks. Under a specific set of assumptions, it is derived that the SLIDE algorithm keeps the correlation of the watermarks proportional to those of the host frames up to a certain resolution related to the total number of anchor points in a frame.

It should be noted that overall digital fingerprinting is a passive form of security and works only *after* the content has been received and made available to the user. In the next section, we discuss video encryption, an active form of protection that when applied prevents a user from content access.

IV. Video Encryption

A. Partial Encryption

Encryption can be defined as a transformation that is parameterized by a numeric value called an *encryption key* K_E of a given input signal called the *plaintext*. The output of the transformation, called the *ciphertext*, must ideally "appear" random to make estimation of the plaintext from the ciphertext computationally difficult without access to the decryption key K_D; K_D may be the same or different from K_E depending on the type of transformation employed. The process of decryption is the inverse transformation of encryption.

Video encryption has gained interest in recent years because use of well-known and tested secret key encryption algorithms, such as Triple Data Encryption Standard (3DES) and Advanced Encryption Standard (AES), are considered computationally infeasible for high volumes of plaintext in low-complexity devices or for near real-time or massively parallel distribution of video flows. In order to overcome some limitations, Cheng and Li [24] discuss the notion of *partial encryption*, in which a smaller subset of the plaintext is encrypted to lower computation and delay while integrating the overall process with compression.[2] Fig. 4 summaries the basic idea. The raw video data X is partially compressed (e.g., the transform coefficients of X are quantized) and then separated into two components: the essential features (EF) X_{EF} that must be encrypted and the nonessential features (NEF) X_{NEF} that are left in plaintext form. The output of the encryption stage denoted $E_{K_E}[X_{\mathrm{EF}}]$ is further compressed (which involves nonlossy compression such as entropy coding, for example) and the result is a multiplexed with X_{NEF} to produce the final encrypted and compressed content.

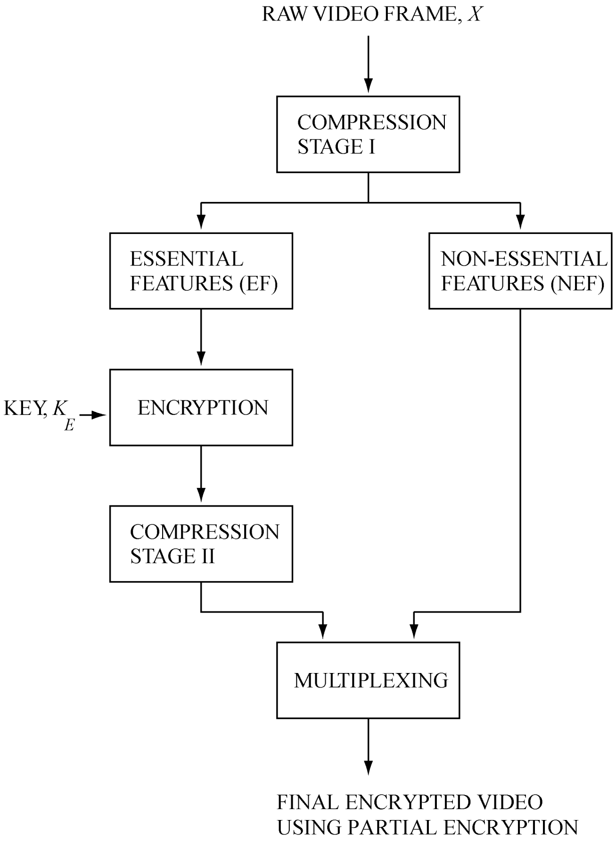

Fig. 4. Overview of partial encryption of raw video.

Using this partial encryption framework, the authors of this paper assert that the problem of designing an effective video encryption algorithm involves selecting the appropriate EF and NEF of the video content for a given application. The EF–NEF selection may be based on a number of different criteria.

1) It is desirable that the fraction of the video stream that needs to be encrypted is as small as possible; thus,

[2] Another effort to address this problem, which we do not address in this paper, involves the digital video broadcasting (DVB) scrambling system suitable for set-top boxes.

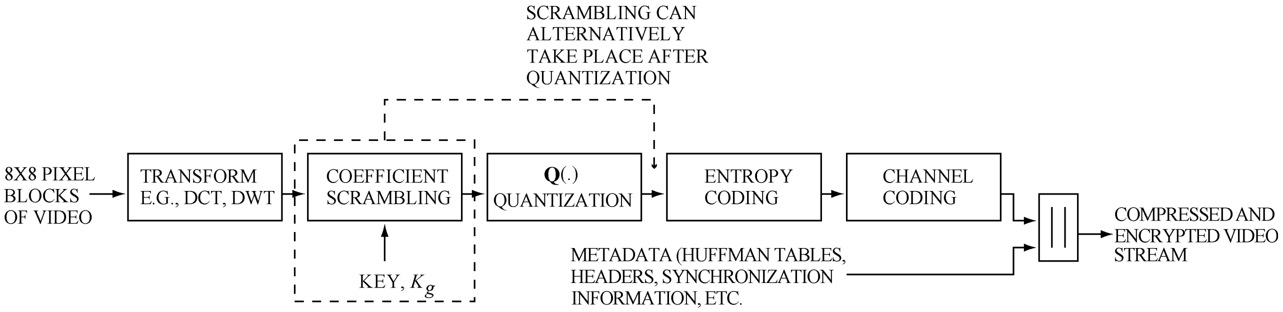
SCRAMBLING CAN ALTERNATIVELY TAKE PLACE AFTER QUANTIZATION
8X8 PIXEL BLOCKS OF VIDEO
TRANSFORM E.G., DCT, DWT
COEFFICIENT SCRAMBLING
KEY, K_g
$\mathbf{Q}(.)$ QUANTIZATION
ENTROPY CODING
CHANNEL CODING
METADATA (HUFFMAN TABLES, HEADERS, SYNCHRONIZATION INFORMATION, ETC.
COMPRESSED AND ENCRYPTED VIDEO STREAM

Fig. 5. Video encryption by coefficient and sign scrambling.

the bit rate of X_{EF} should ideally be much smaller than the bit rate of the X_{NEF} after the first stage of compression.

2) The security in a video watermarking context is related, in part, to visual quality. Thus, the EF–NEF partitioning should guarantee that the "visual quality" of the encrypted video is highly dissimilar to the plaintext. That is, X_{EF} should contain most of the perceptually critical content of X. In some applications, only the commercial quality of the signal needs to be degraded by scrambling allowing some of the original content through.
3) It should not be possible for an attacker to estimate the EF from the NEF. Otherwise, it may be possible to obtain $\hat{X}_{\mathrm{EF}}$, an estimate of X_{EF} from X_{NEF} and then use $E_{K_E}[X_{\mathrm{EF}}]$ and $\hat{X}_{\mathrm{EF}}$ in a traditional cryptographic known-plaintext attack.
4) If the compression stages are based on a coding standard, then the X_{EF} and X_{NEF} components must be easily accessible elements related to the compression process, such as discrete cosine transform (DCT) coefficients in the case of MPEG-2.

B. Practical Considerations for a Video Encryption Scheme

In addition to the complexity and partial encryption considerations of the previous sections, we highlight other practical requirements in video encryption design. Communication latency, which refers to the time lag between the source of communication and its destination, must be minimized. This is a more serious issue if the encrypted transmission has to be done in real time for applications such as video conferencing as opposed to VoD; however, severe latency is a problem for, say, live Web broadcasts. Friedman [25] demonstrates through simulations that using the IP Sec protocol with encryption and authentication using 3DES and SHA-1, respectively, for securing real-time voice communications introduced an end-to-end delay of 1.2 ms; the associated delays for video would be unacceptable.

In addition, transcodability must be possible if customers in a VoD system have different QoS constraints. Transcoding is the process of converting directly from one video format to another with often lower quality requirements to facilitate content distribution to users with varying bandwidths. Practically, transcoders can be found at video gateways that connect fast networks to slow ones. A straightforward approach to integrating transcoding with encryption requires a complete decryption of the video content, then standard transcoding, and finally reencryption. Chang *et al.* [26] and Wee and Apostolopoulos [27] propose subdividing the video content into multiple components, each independently encrypted such that the transcoder can judiciously drop some of the encrypted components without processing them and yet not affecting the decryption of the remaining ones.

Finally, the influence of encryption on the video compression rate must be minimal. For overall efficiency, encryption and compression are often integrated as discussed in Fig. 4 where compression is broken up into two stages. Attempts of compression at Stage II (after the encryption) are often ineffective because the scrambling increases entropy such that further compression has diminishing returns. Thus, the compression stages should be designed such that much information reduction occurs in Stage I while leaving the data in a form where X_{EF} and X_{NEF} with encryption-friendly characteristics can be easily separated and processed accordingly.

C. Existing Partial Encryption Approaches

Many of the techniques for partial encryption are proposed for efficient MPEG encryption. Therefore, intermediate signal elements related to the MPEG standards, such as the DCT or discrete wavelet transform (DWT) coefficients, are convenient to use as X_{EF} and X_{NEF}.

1) Coefficient Scrambling: The coefficient scrambling class of methods encrypts some property of the video signal such as the overall value, position or sign of its DCT or DWT coefficients (depending on the codec used for compression). As Fig. 5 demonstrates, the scrambling occurs either immediately before or after quantization. The advantages include low computational complexity and easy integration with compression. The primary disadvantage is that this approach is not necessarily secure if simple shuffling procedures are used to obscure the coefficients.

In [28] Tang proposes a DCT shuffling scheme for compatibility with JPEG or MPEG-2, in which the DCT coefficients within each 8 × 8 block are permuted using a secret key. Zigzag scanning has the same computation order as shuffling, so no computational security overhead over compression is incurred. The scheme, although simple to implement, changes the statistical (run length) property of the DCT coefficients, making it less susceptible to entropy coding and, therefore, increases the bit rate of the encrypted stream. This problem, however, can be alleviated by restricting the scrambling to the lower DCT frequencies (that contains most of the signal energy) without significantly compromising on security.

To overcome the increased bit rate, Shi and Bhargava [29] propose the MPEG Video Encryption Algorithm (VEA) that encrypts the sign bits of all the DCT coefficients[3] of an MPEG video stream. The main advantage is that VEA adds minimum overhead to the MPEG codec, because encryption involves a bitwise XOR operation of each nonzero DCT coefficient with the secret key.

2) MPEG Bit Stream Encryption: For increased security over the shuffling procedures of the previous section, a number of partial encryption proposals use algorithms such as DES, 3DES, and AES. However, to sustain the same level of complexity, a lower volume of video must be encrypted. The challenge with this method is that the sparse encrypted components may be treated by an attacker as "errors" in the transmitted video stream that can be corrected by using the natural redundancy in the video stream. Such error correction capability is equivalent to breaking the partial encryption algorithm. Thus, care must be taken to select V_{EF} such that it contains critical information that cannot be estimated from V_{NEF}

Early attempts suggest encrypting only the I-frames or Intra-blocks of the MPEG video stream. However, Agi and Gong [30] show that although it may appear that the P- or B-frames are visually meaningless without the corresponding I-frame, a series of P- and B- frames carries much more perceptual information especially if their base I-frames are correlated, which can be used to increase fidelity from the encrypted video stream. One solution is to increase the frequency of the I-frames and, hence, encrypted content, which in turn raises the bit rate. The proposal by Agi and Gong involves using DES to encrypt the I-frames. However, one problem with the use of block ciphers is that an error in a single bit will render the entire decrypted block unintelligible.

Qiao and Nahrstedt [31] proposed a video encryption scheme that works exclusively on the data bytes and does not interpret the MPEG stream for selection of V_{EF}. In this algorithm the authors divide the MPEG stream into two components composed of odd- and even-numbered bytes. Through statistical analysis, they show that there is no repetitive pattern in the even byte stream which has random characteristics similar to that desirable for an encryption key. Using the even byte stream as a key for a one-time pad encryption algorithm, the odd numbered bytes are XORed with the even ones. To protect the identity of the even stream, it is then protected by applying a standard algorithm such as DES. The result is a 47% reduction in computation as compared to the full encryption scheme.

3) Hierarchically Based Encryption: In compression schemes based on multiresolution analysis of images such as quadtree decomposition or zerotrees wavelet compression, there is a correlation between sets of coefficients that exist at various levels of the tree. Coefficients that have similar characteristics are grouped together by forming linked lists. Since each linked list is essentially a chain of pointers, one can ensure secrecy by encrypting just the *parent set* (or node) and the corresponding parameters that describe the parent set, so that the subsequent links lack reference information and are ideally useless in deciphering the video content.

Using such an approach, Cheng and Li [24] and Shapiro [32] propose partial encryption for quadtree compression and wavelet compression based on zerotrees, respectively. The main advantage is that only 13%–27% and 2% of the compressed output of typical images need to be compressed for [24] and [32], respectively.

V. Joint Fingerprinting and Decryption

As discussed in Section II-C, we propose a new architecture for both tracing pirates and securing multimedia across multicast networks, in which we combine the process of fingerprinting and decryption at the receiver. Shifting the fingerprinting process to the receiver and, thus, integrating it with decryption has the advantage that the media needs to be encrypted just once at the source before being multicast to the different receivers, a tremendous saving in computational power, memory, latency, and bandwidth, over architectures discussed in Sections II-A and II-D without requiring tamperproofing hardware.

The discussions in Sections III and IV highlight a number of important compromises for multimedia security design. In this section, we investigate possible advantages of using strategies to integrate on one or more processing stages. Specifically, we propose a new architecture for both tracing pirates and securing multimedia across multicast networks, in which we combine the process of fingerprinting and decryption at the receiver as discussed in Section II-C. Shifting the fingerprinting process to the receiver has the advantage that the media needs to be encrypted just once at the source before being multicast to the different receivers, a tremendous saving in computational power, memory, latency, and bandwidth, over architectures discussed in Sections II-A and II-D. Since every receiver can be a potential pirate, the fingerprint embedding process must be made inaccessible. The JFD architecture is a more computationally efficient and convenient way of fingerprinting at the receiver without the need for expensive tamperproofing equipment.

A. Related Work

Our architecture is inspired by the work of Anderson and Manifavas on *chameleon ciphers* [33] developed for multicast- or broadcast-type channels. Chameleon ciphers are a clever way to combine encryption and fingerprinting that offers computational efficiency. The idea is that encryption is performed on the plaintext at the source using a group key K_S to produce the ciphertext that is multicast to all the n users. Slightly different decryption keys from the set $K_R = \{K_1, K_2, \ldots, K_n\}$ are distributed to the users such that when decryption is performed on the ciphertext, the

[3] More exactly, the differential DC values of the DC coefficients are encrypted.

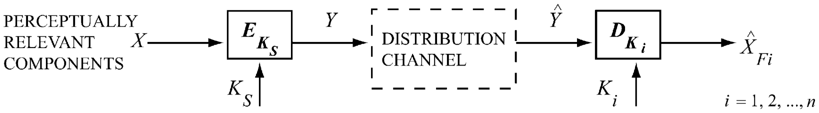

Fig. 6. Overview of the JFD architecture.

result is slightly different for each user. In their implementation for raw audio streams, Anderson and Manifavas adapt a block cipher for encryption in output feedback mode to operate on raw audio pulse code modulation bits and ensure that the least significant bits of the plaintext audio alone are changed when the content is decrypted with a user's decryption key. This guarantees that the fingerprint does not result in perceptual degradation. Some of the challenges involving chameleon ciphers include robustness of the fingerprint due to its LSB nature, the large key size that is inappropriate for multicast scenarios, the assumption that the media signal is in raw format rather than in compressed form, and its vulnerability to collusion. It is shown through simulations that five or more users can produce a plaintext that cannot be traced by using a bit-wise majority voting to erase the watermark from the LSBs.

More recently, a scheme by Parvianen and Parnes [34] has been proposed in which the authors also adapt a stream cipher such that different users equipped with long and distinct decryption keys receive fingerprinted copies of the plaintext. Their algorithm allows the use of more general fingerprinting techniques because the process of embedding occurs separately from encryption at the source, but in such a way that the bandwidth usage is at most doubled over that of normal multicast. Analysis is provided to demonstrate that if colluders represent small fractions of the overall users in the system, the detection likelihood of traitors is high.

B. JFD Architecture

The source extracts the perceptually relevant features from the multimedia content X and selectively encrypts them with K_S as shown in Fig. 6. Based on this model, the source multicasts the encrypted content Y to $(n > 1)$ users. Each receiver upon subscription receives a decryption key K_i from the source. The decryption key set denoted $K_R = \{K_1, K_2, \ldots, K_n\}$ is designed jointly with the source key set K_S so that an imperceptible and indelible fingerprint is embedded in the content after decryption. The reader should note that although we require $K_i \neq K_S$, we do not employ asymmetric encryption. The key asymmetry, as it will become clear, stems from the necessity of jointly decryption and fingerprint embedding.

For user i, the fingerprint information is essentially contained in the asymmetric key pair (K_S, K_i) out of which the ith receiver has access only to the decryption key K_i and the encrypted content $\hat{Y} = E_{K_S}[X]$. The receivers do not have knowledge of K_S. The fingerprint is a function of the correlation between the encryption and decryption keys. We assume the encryption and decryption keys are correlated random variables and an encryption and decryption structure with limited *diffusion* capability is employed; the source of the secrecy comes primarily from the process of *confusion* [35]. We define the fingerprint payload capacity, which is the maximum length of the fingerprint that can be embedded in a given media object, as

$$C_{F_i} \triangleq H(K_S|K_i) = H(K_S) - I(K_S; K_i) \qquad (2)$$

where $H(K_S)$, $I(K_S; K_i)$, and $H(K_S|K_i)$ are the entropy of the encryption key, the mutual information between the encryption and decryption keys, and the conditional entropy of the encryption key given the decryption key, respectively. Since the fingerprint embedding is done within the decryption framework, the perceptual quality of the fingerprinted frame can be described as a function of the correlation between K_S and K_i. The mutual information $I(K_S; K_i)$ is a function of the correlation between the keys, which in turn affects the perceptual quality of the decrypted/fingerprinted image given our encryption structure. Intuitively one can say that the higher the correlation, the smaller the perceptual degradation. However, it is also clear that the larger the value of $I(K_S; K_i)$ due to the increased correlation, the lower the fingerprint payload capacity C_{F_i}. Thus, (2) illustrates the inherent tradeoff between fingerprint payload capacity and perceptual quality as a byproduct of combining watermarking and decryption.

C. Design Challenges

Many interesting challenges arise in the JFD framework owing to the merging of the two seemingly orthogonal processes of watermarking and encryption. Traditional fingerprinting leaves a triangular tradeoff among robustness, imperceptibility, and capacity. However, the JFD framework has an additional fourth element of secrecy defined as the strength of an encryption. Our focus in this paper is on issues related to fingerprint design within a JFD framework, so we do not discuss security extensively in the remainder of this section, which is the topic of another paper.

1) Imperceptibility Versus Secrecy: Imperceptibility of the fingerprint after decryption with K_i is necessary to ensure that users with valid decryption keys are able to decrypt the content properly without degrading the perceptual quality. Given a key K_D, which need not be in the key space K_R (since K_D may not be a legitimate decryption key, but the result of collusion), two necessary conditions for the asymmetric fingerprinting scheme are

$$\text{if } K_D \in K_R \text{ then } P_D(\hat{X}_{Fi}, X) < \delta_{p1}, \quad \text{for } i = 1, 2, \ldots, n \qquad (3)$$

$$\text{else if } K_D \notin K_R \text{ then } P_D(\hat{X}_{Fi}, X) > \delta_{p2}, \quad \text{for } i = 1, 2, \ldots, n \qquad (4)$$

with high probability

where $P_D(\cdot,\cdot)$ is the perceptual distortion measure, δ_{p1} is a global masking threshold for all the n fingerprints, and δ_{p2} is the minimum perceptual distance necessary to obscure the viewing experience if the wrong key is used. Condition (3) guarantees that a legitimate user can decrypt the content with reasonable perceptual quality, and condition (4) suggests that if the wrong key is used, there is a severe degradation of the media quality; thus, content access is denied. The distortion measure $P_D(\cdot,\cdot)$ is an application- and media-dependent quantity; contenders for this metric are the L^1 and L^2 norms and peak signal-to-noise ratio (PSNR) measure.

2) Collusion Attack Model and Countermeasures: We perceive two types of collusion attacks in this framework: 1) collusion of decryption keys, which is a protocol attack [Type (A)], and 2) collusion of the fingerprinted video frames, which is a signal processing attack [Type (B)]. The fingerprint must be robust to both types of attacks. We focus on Type (B) collusion and restrict the attack to be linear in this preliminary formulation.

Definition 1 [Type (A): Collusion of Decryption Keys]: We can break up the set of receivers R into two mutually exclusive sets: colluders C and innocent users U such that $R = C \cup U$. Correspondingly, we can split up the decryption key space into two disjoint spaces, such that $K_R = K(C) \cup K(U)$, where $K(C) = \{K_1^C, K_2^C, \ldots, K_t^C\}$. These unknown t colluders in the set C, can collude the keys using some function Φ to obtain an estimate of a decryption key as follows:

$$\hat{K}_C = \Phi\left(K_1^C, K_2^C, \ldots, K_t^C\right). \tag{5}$$

The nature of the function Φ depends specifically on the structure of the keys and the type of encryption used.

Definition 2 (Attack—Framing): If constraints (3) and (4) are met successfully by the key designer, then the pirates will have no choice but to choose the function Φ such that $\hat{K}_C \in U$, i.e., the key is within the decryption key space of innocent users. This attack is called framing.

Proposition 1 (Frameproof Coding): The decryption keys must be designed in such a way that for any computationally possible function Φ, the key estimate $\hat{K}_C$: 1) must not map to one of the keys in the decryption key space K_R or 2) must map to one of the colluders.

Definition 3 [Type (B): Linear Collusion of Fingerprinted Copies]: Given $\hat{X}_{F1}, \hat{X}_{F2}, \ldots, \hat{X}_{Ft}$, the set of fingerprinted copies available to the t colluders, the colluded copy is generated as follows:

$$\hat{X}_{Fc} = \frac{\hat{X}_{F1} + \hat{X}_{F2} + \ldots + \hat{X}_{Ft}}{t}. \tag{6}$$

We assert that one way to mitigate the effect of a large-scale linear collusion is by introducing some common elements (or invariant marks) in the fingerprints distributed to subsets of users. In this paper we propose a scheme in which any combination of t or fewer colluding groups out of n can be detected with high probability. In this approach, which can be considered a variation of the Dittmann *et al.* scheme [20], for each unique combination of colluders, a different set of marks (or bits) are preserved that can be used to determine the exact colluding members.

D. Joint Fingerprint Embedding and Decryption Based on Coefficient Set Scrambling

To demonstrate the practical compromises necessary for algorithm design, we implement a straightforward JFD scheme based on scrambling of the DCT coefficients. For simplicity, we focus on results for a single image rather than a video stream, but the same insights hold for both types of media.

1) Fingerprint Embedding: Our scheme, in part, may be considered a merging of the method by Tang [28] with the notion of chameleon ciphers by Anderson and Manifavas [33] to generate a set of fingerprints during decryption with the invariant character discussed by Dittmann *et al.* scheme [20]. The raw video frame is DCT encoded and quantized. The DCT is taken of the entire image, and it is not divided into 8 × 8 blocks as in the case of [28] for reasons of perceptibility of fingerprint during decryption. We then identify a set of coefficients in the low and midfrequency region that are perceptually significant. From this set, we partition the coefficients into n subsets. These subsets are all sign scrambled (i.e., the sign is arbitrarily flipped depending on the key) during encryption and some are left scrambled during decryption for the purpose of forming the fingerprint as in the case of [33].

There are two types of encryption keys associated with each component; the first serving as pointers to subsets of coefficients to encrypt, and the second as the scrambling key that dictates the order in which the coefficients are permuted within a given subset. The receiver is given a unique subset of keys K_i for decrypting only a p/n fraction of the n encrypted subsets. The remaining $k = n - p$ subsets are hidden from the receiver and their unique sign bit signature along with their concealed positions in the frame constitutes the fingerprint. An overview of the system is provided in Fig. 7. The length of the receiver key set K_i in relation to the source key set is a measure of the correlation between the two keys, since $K_i \subset K_S$.

The number of coefficients in each subset denoted M_j (where j is the subset index) is chosen to trade perceptual quality and robustness. As M_j increases, there is a corresponding increase in diversity of the fingerprint embedded, which, consequently, increases the probability of retrieving the mark. However, if a large number of coefficients are left scrambled, this will increase the perceptual distortion in the fingerprinted frame.

The only way to meet the imperceptibility constraint without compromising on robustness would be by decreasing the number of hidden subsets k, which results in a decrease in the fingerprint payload. Thus, we, once again, have the triangular tradeoff scenario in traditional watermarking.

If k is small as compared to n, then the fingerprints will be almost orthogonal to each other, and a maximum of ${}^nC_k$

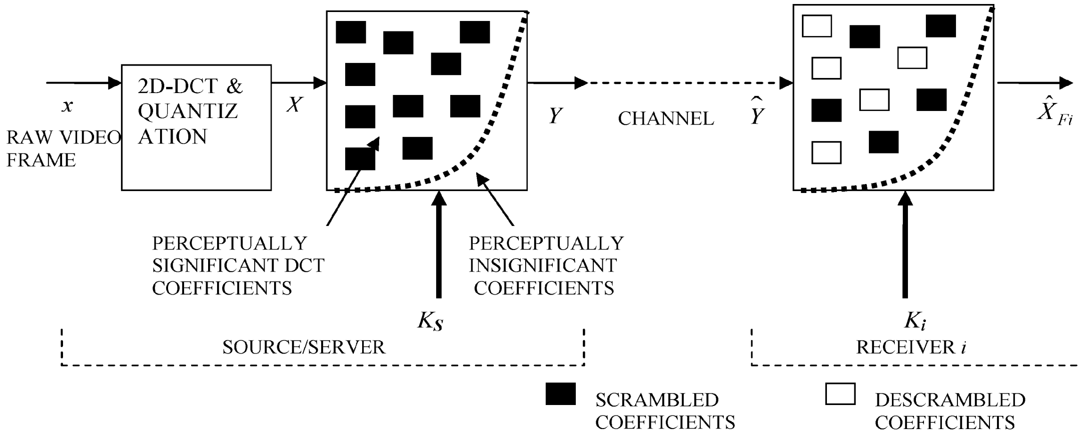

Fig. 7. Overview of fingerprint embedding through subset scrambling.

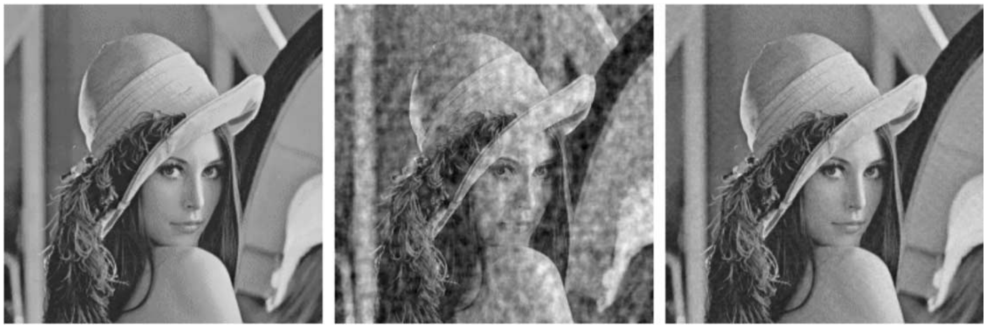

Fig. 8. Original, encrypted, and fingerprinted images (for $n = 25, k = 6, M_B = 200$).

Table 1
Results Demonstrating Tradeoff Between Payload, Robustness, and Perceptual Quality

	$k = 5$ $P_e, PSNR(\hat{X}_{Fi})$	$k = 7$ $P_e, PSNR(\hat{X}_{Fi})$(dB)	$k = 10$ $P_e, PSNR(\hat{X}_{Fi})$	$PSNR(Y)$(dB)
$n = 25, M_B = 100$	$1/5, 41$	$1/7, 37$	$1/10, 34$	26
$n = 25, M_B = 200$	$1/5, 38$	$1/7, 36$	$1/10, 29$	22
$n = 25, M_B = 300$	$1/5, 33$	$2/7, 33$	$1/10, 28$	21
$n = 25, M_B = 400$	$2/5, 31$	$2/7, 31$	$2/10, 26$	20

fingerprints can be embedded. But as the number of colluders increases in an orthogonal fingerprinting scheme, the false positive and the false negative rates increase, mainly because the distinguishing features between the fingerprints assume negligible amplitude when the copies are averaged. If we introduce some "common" elements among subsets of colluders, then these elements are likely to be preserved with a higher probability. However, there is a sacrifice in resolution for reliable detection. This lays the foundation for a group-based fingerprinting scheme by introducing a two-level hierarchical structure to fingerprinting. Thus, in the present scheme, the fingerprint has two components, the group ID and the user ID.

2) Fingerprint Detection and Simulation Results: Each fingerprint resulting from partial decryption has a unique *sign bit signature*. Fingerprint detection is carried out by correlating the subsets in the retrieved copy with a list of reference signatures. Let $S_{\mathrm{REF,n}} = [\bar{s}_1 \bar{s}_2 \ldots \bar{s}_n]^T$ be the sign signatures of the n subsets in the encrypted frame Y and $\hat{X}_{\mathrm{test}}$ be the retrieved copy from which the colluders (if any) have to be detected. In a similar fashion, we create a sign matrix $S_{\mathrm{TEST,n}}$ from $\hat{X}_{\mathrm{test}}$, where $S_{\mathrm{TEST,n}} = [\hat{s}_1 \hat{s}_2 \ldots \hat{s}_n]^T$. We then form the correlation matrix $R = (1/M_B)[S_{\mathrm{REF,n}} \cdot S^T_{\mathrm{TEST,n}}]$ and an autocorrelation vector, $C = |diag(R)| = [\rho_1 \rho_2 \cdots \rho_n]^T$, $0 \leq \rho_i \leq 1$. M_B is the average number of coefficients in each subset. If $\rho_i \geq T_i$, then we declare that the subset is encrypted and the corresponding video frame component is marked as "1"; otherwise, it is marked as "0." The locations of the ones in the string constitute the fingerprint, which is compared with the entries in the database to identify potential colluders. The threshold is a function of the design parameters n, k, M_B and can be adjusted experimentally to balance the false positive and false negative rates.

For a 256 × 256 grayscale Lena image, we set the parameters to be $n = 25$, $k = 6$, $M_B = 200$. The corresponding PSNR values of the encrypted and fingerprinted images were found to be $\delta_{p1} = 22$ dB and $\delta_{p2} = 34$ dB, respectively. We set these as the lower and upper bounds for obscurity (or secrecy) and imperceptibility respectively (better bounds can be obtained experimentally by averaging the PSNR values over an ensemble of images with different textures). Fig. 8 shows the original, scrambled, and fingerprinted images. Our intent is only to obscure the commercial quality of the video.

Table 1 illustrates the tradeoff between imperceptibility PSNR($\hat{X}_{\mathrm{Fi}}$), robustness, and payload (${}^nC_k$). A JPEG compression ($Q = 60$) has been applied to the fingerprinted image. We measure robustness as the fraction of hidden marks (P_e) inaccurately detected. Payload can be increased by increasing k or n and diversity by increasing M_B. As seen from the table, increasing both k and M_B has a severe impact on the perceptual quality of the fingerprinted image. To maintain a minimum level of secrecy ($\text{PSNR}(Y) \leq \delta_{p1} = 22$ dB) without compromising on robustness, the width of the subset must be greater than 100. Beyond a particular value of M_B, the error rate surprisingly increases because of overlapping subsets, which causes false negatives.

E. Work in Progress and Future Challenges

The preliminary scheme discussed here was developed heuristically and later adapted to meet design rules to illustrate the feasibility of the JFD approach. Although we have proposed ways of tracing subsets of colluders in the face of linear collusion attacks, this scheme is susceptible to key collusion attack. A group of pirates can compare the key sets to successfully identify and erase some of the subbands that have been hidden in the frame. However, framing is very difficult, since accurate reconstruction of an innocent user's fingerprint would require exact knowledge of his/her decryption key, which is improbable.

The receiver should not know the video features being decrypted nor the features hidden; in the presented scheme, the former is available to the receiver. Work in progress investigates designing a system such that the receiver is not able to derive information about the location of encrypted components without access to the source secret key. At present, we are also working on an analytical model for illustrating tradeoff issues in the JFD framework, which can then be used to develop a more robust fingerprinting scheme.

VI. Conclusion

This paper, in part, provides an overview of the many issues that must be addressed for video encryption and fingerprinting in a DRM context. Given the thrust toward security for emerging resource constrained DRM applications, there is a need for solutions that provide a better compromise between security and complexity. This is resulting in a paradigm shift in the area of information protection, in which ideas from areas such as media processing are often incorporated to provide more lightweight solutions.

Low-complexity security solutions must take careful account of the application-dependent restrictions and competing objectives. Current solutions often strip down the security algorithm or protect a partial component of the information. Inspired by the chameleon cipher, we focus on the approach of integration to potentially achieve a more appropriate compromise. Given that perfect security may be unattainable in a practical context, we hope that further research in this area will result in more effective yet efficient designs.

References

[1] N. R. Wagner, "Fingerprinting," in *Proc. IEEE Symp. Security and Privacy*, 1983, pp. 18–22.

[2] F. Hartung and B. Girod, "Digital watermarking of MPEG-2 coded video in the bitstream domain," in *Proc. Int. Conf. Acoustics, Speech and Signal Processing*, vol. 4, 1997, pp. 2621–2624.

[3] D. Boneh and J. Shaw, "Collusion-secure fingerprinting for digital data," *IEEE Trans. Inform. Theory*, vol. 44, pp. 1897–1905, Sept. 1998.

[4] W. Trappe, M. Wu, and K. J. R. Liu, "Collusion-resistant fingerprinting for multimedia," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, vol. 4, 2002, pp. 3309–3312.

[5] W. Trappe, M. Wu, Z. J. Wang, and K. J. R. Liu, "Anti-collusion fingerprinting for multimedia," *IEEE Trans. Signal Processing*, vol. 51, pp. 1069–1087, Apr. 2003.

[6] B. M. Macq and J.-J. Quisquater, "Cryptology for digital TV broadcasting," *Proc. IEEE*, vol. 83, pp. 944–957, June 1995.

[7] J. Bloom, "Security and rights management in digital cinema," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, vol. 4, Apr. 2003, pp. 712–715.

[8] W. Luh and D. Kundur, "Media fingerprinting techniques and trends," in *Multimedia Security Handbook*, B. Furht and D. Kirovski, Eds. Boca Raton, FL: CRC, 2004, ch. 19.

[9] R. Anderson and F. A. P. Petitcolas, "On the limits of steganography," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 474–481, May 1998.

[10] I. J. Cox and M. L. Miller, "A review of watermarking and the importance of perceptual modeling," *Proc. SPIE, Human Vision and Electronic Imaging II*, vol. 3016, pp. 92–99, 1997.

[11] G. R. Blakely, C. Meadows, and G. B. Purdy, "Fingerprinting long forgiving messages," in *Proc. Advances in Cryptology*, 1985, pp. 180–189.

[12] B. Pfitzmann and M. Schunter, "Asymmetric fingerprinting," in *Proc. Eurocrypt 1996*, pp. 84–95.

[13] B. Pfitzmann and M. Waidner, "Anonymous fingerprinting," in *Proc. Eurocrypt 1997*, pp. 88–102.

[14] S. Baudry, J. F. Delaigle, B. Sankur, B. Macq, and H. Maitre, "Analyses of error correction strategies for typical communication channels in watermarking," *Signal Process.*, vol. 81, no. 6, pp. 1239–1250, June 2001.

[15] P. Fernando, P. Gonzalez, J. R. Hernandez, and F. Balado, "Approaching the capacity limit in image watermarking: a perspective on coding techniques for data hiding applications," *Signal Process.*, vol. 81, no. 6, pp. 1215–1238, June 2001.

[16] B. Chen and G. W. Wornell, "Quantization index modulation: A class of provably good methods for digital watermarking and information embedding," *IEEE Trans. Inform. Theory*, vol. 47, pp. 1423–1443, May 2001.

[17] D. Kundur and D. Hatzinakos, "Diversity and attack characterization for improved robust watermarking," *IEEE Trans. Signal Processing*, vol. 29, pp. 2383–2396, Oct. 2001.

[18] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Trans. Image Processing*, vol. 6, pp. 1673–1687, Dec. 1997.

[19] H. Zhao, M. Wu, Z. J. Wang, and K. J. R. Liu, "Nonlinear collusion attacks on independent fingerprints for multimedia," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 5, 2003, pp. 664–667.

[20] J. Dittmann, A. Behr, M. Stabenau, P. Schmitt, J. Schwenk, and J. Ueberberg, "Combining digital watermarks and collusion secure fingerprints for digital images," in *In Proc. SPIE Conf. Electronic Imaging '99, Security and Watermarking of Multimedia Contents*, vol. 41, 1999, pp. 171–182.

[21] K. Su, D. Kundur, and D. Hatzinakos, "A novel approach to collusion-resistant video watermarking," in *Proc. SPIE, Security and Watermarking of Multimedia Content IV*, vol. 4675, 2002, pp. 491–502.

[22] ——, "Statistical invisibility for collusion-resistant digital video watermarking," *IEEE Trans. Multimedia*, to be published.

[23] ——, "Spatially localized image-dependent watermarking for statistical invisibility and collusion resistance," *IEEE Trans. Multimedia*, to be published.

[24] H. Cheng and X. Li, "Partial encryption of compressed images and videos," *IEEE Trans. Signal Processing*, vol. 48, pp. 2439–2451, Aug. 2000.

[25] A. Friedman, "Wireless and mobile communication: A real time security solution over wireless IP," Fortress Technol., White Paper, 1999.

[26] Y. Chang, R. Han, C. Li, and J. Smith, "Secure transcoding of internet content," in *Proc. Int. Workshop Intelligent Multimedia Computing and Networking (IMMCN)*, 2002, pp. 940–943.

[27] S. J. Wee and J. G. Apostolopoulos, "Secure scalable streaming enabling transcoding without decryption," in *Proc. IEEE Int. Conf. Image Processing*, vol. 1, 2001, pp. 437–440.

[28] L.Lei Tang, "Methods for encrypting and decrypting MPEG video data efficiently," in *Proc. 4th ACM Int. Conf. Multimedia*, 1996, pp. 219–229.

[29] C. Shi and B. Bhargava, "A fast MPEG video encryption algorithm," in *Proc. 6th ACM Int. Multimedia Conf.*, 1998, pp. 81–88.

[30] I. Agi and L. Gong, "An empirical study of secure MPEG video transmissions," in *Proc. Internet Society Symp. Network and Distributed System Security*, 1996, pp. 137–144.

[31] L. Qiao and K. Nahrstedt, "Comparison of MPEG encryption algorithms," *Comput. Graph.*, vol. 22, no. 4, pp. 437–448, 1998.

[32] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 41, pp. 3445–3462, Dec. 1993.

[33] R. Anderson and C. Manifavas, "Chameleon—A new kind of stream cipher," in *Lecture Notes in Computer Science, Fast Software Encryption*, E. Biham, Ed. Heidelberg, Germany: Springer-Verlag, 1997, pp. 107–113.

[34] R. Parnes and R. Parviainen, "Large scale distributed watermarking of multicast media through encryption," in *Proc. IFIP Int. Conf. Communications and Multimedia Security Issues of the New Century*, 2001, p. 17.

[35] C. E. Shannon, "Communication theory of secrecy systems," *Bell Syst. Tech. J.*, vol. 28, pp. 656–715, Oct. 1949.

Deepa Kundur (Senior Member, IEEE) was born in Toronto, ON, Canada. She received the B.A.Sc., M.A.Sc., and Ph.D. degrees in electrical and computer engineering in 1993, 1995, and 1999, respectively, at the University of Toronto, Toronto.

From 1999 to 2002, she was an Assistant Professor in the Edward S. Rogers Sr. Department of Electrical and Computer Engineering, University of Toronto, where she held the title of Bell Canada Junior Chair-holder in Multimedia. In 2003, she joined the Electrical Engineering Department at Texas A&M University, College Station, where she is a member of the Wireless Communications Laboratory and holds the position of Assistant Professor. She is the author of over 60 papers. Her research interests include multimedia and network security, video cryptography, sensor network security, data hiding and steganography, covert communications, and nonlinear and adaptive information processing algorithms.

Dr. Kundur is the recipient of several awards, including the 2002 Gordon Slemon Teaching of Design Award. She has been on numerous technical program committees and has given over 30 talks in the area of digital rights management, including tutorials at ICME 2003 and Globecom 2003.

Kannan Karthik (Student Member, IEEE) received the B.E degree in electrical engineering from Bombay University in 1998 and the M.Eng degree in Power Electronics and Controls from Memorial University of Newfoundland, St. John's, Canada in 2001. He is currently working toward the Ph.D degree in Multimedia Security at the University of Toronto, ON, Canada.

His research interests include multimedia security, video watermarking and cryptography, and statistical signal processing.