Decryption Key Design for Joint Fingerprinting and Decryption in the Sign Bit Plane for Multicast Content Protection

Kannan Karthik and Dimitrios Hatzinakos

(Corresponding author: Kannan Karthik)

Department of Electrical and Computer Engineering, University of Toronto
Toronto, ON, M5S3G4, Canada. (Email: {karthik, dimitris}@comm.utoronto.ca)

(Received Sept. 28, 2005; revised and accepted Nov. 17, 2005)

Abstract

The prime objective in this paper is to explore the possibility of combining the seemingly orthogonal processes of watermarking and encryption along the spirit of the chameleon cipher. By integrating perceptual models indirectly through a 'weeding' process and collusion resistant coding methodologies in the design of the decryption keys, we have shown that it is possible to embed a robust yet relatively imperceptible fingerprint at the receiver. We have also provided some insights into the softening of the encryption process due to the enforcement of additional perceptual constraints because of fingerprinting.

Keywords: Collusion resistance, joint fingerprinting and decryption (JFD) , key collusion, sign bit embedding.

1 Introduction

FINGERPRINTING in a broad context refers to the art of embedding or concealing imperceptible, transparent and inconspicuous objects or marks in video, images [11], audio [8], documents [4], software [7] and even hardware[12] with the sole objective of tracing any leaks or illicit acts of piracy. Performance measures vary widely based on the type of application. For image and video fingerprinting, which is the focal point of this paper, key challenges lie in minimizing the false positive and negative rates and perceptual distortion (generally measured in terms of peak signal to noise ratio - PSNR) while maximizing the embedded payload. Awareness of perceptual models catered to a particular transform (DCT, wavelets, KLT etc) for signal compression [9] helps in increasing the effectiveness of the fingerprint embedding process.

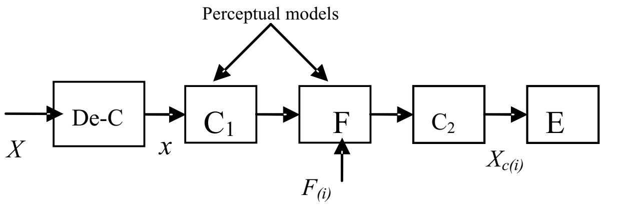
Fingerprinting by itself is only a passive form of protection, which is used to deter piracy. To restrict access to audio and visual information an additional orthogonal protection measure is introduced in the form of encryption. Encryption may be performed before, during or after entropy coding of the quantized coefficients. But the last two methodologies are preferred to the first one because:

1) Encryption processes which are spatial and transform domain based [13, 14], work independently of the signal compression models and are likely to reduce the effectiveness of entropy coding.

2) Removal of spatial-temporal and statistical redundancy is useful pre-processing before applying encryption to selective segments of the stream. This is important from the point of view of cryptanalysis as any media stream is susceptible to error concealment attacks which utilize any residual redundancy in the stream to reconstruct the encrypted segments by treating them as damaged portions.

3) Consider a discrete quantized source which spits independent symbols $\{a, b, c\}$ with probabilities $\{P(a) = 0.5, P(b) = 0.25, P(c) = 0.25\}$ respectively. Such a stream can be Huffman coded as $\{1, 01, 00\}, \{0, 10, 11\}$ or $\{0, 11, 10\}$. The entropy model space increases as the number of symbols increases. Observing the binary stream 10011010100.., does not give the eavesdropper any information about the underlying source coding model (where options abound for large discrete sources such as quantized images and video). Hence, concealment of the model provides sufficient security as synchronization and decoding becomes very difficult without it [9]. This vulnerability of the compressed stream to synchronization errors can be favorably used, as only a fraction of the compressed stream needs to be encrypted if the encryption process is executed during or after entropy coding.

Thus the requirements for a good media fingerprinting and an effective encryption process need not overlap. This obviously challenges our proposed architecture, which is



C_1	Represents transform coding (and) quantization.
C_2	Entropy coding - Huffman or arithmetic coding
E	Encryption process, which can be integrated with C_2 [9] or can be a standard block or stream cipher which operates on selective portions of the compressed stream.
De-C	Decompression process.

Figure 1: Encryption and fingerprinting decoupled

based on joint fingerprinting and decryption (JFD) (inspired by the chameleon cipher [1]). A question that arises is: why should we merge the two processes when they are comfortable in their respective places? The motive for the merger stems from the need for content protection in multicast applications. In Section 2, implementation challenges and some interesting architectures, which provide a partial solution to the multicast security problem, are discussed. Some fundamental design constraints for JFD are presented in Section 3. In Section 4, we present our simple algorithm based on sign bit encryption of images. Analysis and simulation results are presented in Sections 5 and 6 respectively.

2 Multicast Security Architectures

Figure 1 shows an abstraction of the fingerprinting and encryption process in which the latter follows entropy coding ($C2$). It is practical to assume that X represents a compressed image or video stream stored in a protected database, which will be partially or fully decompressed for fingerprinting and then recompressed to $X_{C(i)}$ before being encrypted and then transmitted. The channel coding stage is not shown here as it is symmetric and can be omitted from the abstraction.

Consider the following cases:

1) A single compressed stream X is securely transmitted to various receivers $R = \{U_1, U_2, \ldots U_N\}$.

2) X is fingerprinted and securely transmitted to receivers R.

In Case 1, X is encrypted using encryption key K_E and the encrypted stream $Y = E(X, K_E)$, is multicast to all the receivers in R. Layered encryption techniques can be used to make the stream transcodable without complete decryption which is useful if the receivers have heterogeneous connections.

In Case 2, one may observe that the concept of fingerprinting and multicast appear counterintuitive. Fingerprinting associates a unique identity with each copy of X, which is encrypted and sent to a receiver U_i. Thus N source encryptions are required. This is obviously inefficient from the point of view of bandwidth and computational complexity. In this paper we will focus on our proposed architecture on Joint fingerprinting and decryption (JFD) in which the fingerprinting process is translated to the receivers by merging the keys used for fingerprint insertion and partial stream decryption. In this approach, we design the decryption keys K_{Di}, for a particular lightweight encryption process E such that the decrypted copy $X_{Fi} = D[Y, K_{Di}]$ carries a fingerprint and is perceptually similar to X.

2.1 Multicast Security Requirements

Commercial multicast applications typically have the following characteristics: 1) large number of users; 2) heterogeneous user connectivity (i.e. differential bandwidth demands which require stream transcodability) and 3) dynamic connections (subscribers join and leave the group frequently which call for frequent key refreshes).

If fingerprinting and encryption is done separately, then one can adapt the embedding process carefully based on the perceptual characteristics of the audio and video signal. It also becomes convenient to post-process the watermarking payload with error correction codes (ECC) [2], collusion resistant coding schemes (ACC) [3, 17] and traitor tracing approaches [6]. This independence also helps us design the partial encryption process in such a way that source coding efficiency is unaffected and the encrypted stream is less vulnerable to error concealment attacks.

Note here that, embedding the fingerprint at the source followed by encryption is feasible only for a small user space, as the number of streams fingerprinted, encrypted and unicast scales linearly as the number of users in the group.

2.2 Distributed Watermarking

Distributed watermarking (DW) schemes such as the WHIM [10] and Watercast [5] were proposed to mitigate this bandwidth scalability issue, by incorporating special functions in the multicast routers. These "active" routers were allowed to peck the sequence of packets which were multicast by the source and carve audit trails for tracking the subscribers. With the help of additional logging requirements and increased multicast router complexity the following goals were met: 1) Error resiliency; 2) Lower packet congestion (bandwidth efficiency) and 3) Distributed computation.

These architectures albeit innovative, had some drawbacks. The WHIM architecture comprised of a set of secure servers which performed the dual role of multicast routing and partial watermark embedding thereby creating an audit trail till the last hop where the user IDs were embedded. So the traffic was multicast upto the last hop. The main problem with WHIM was that each active router was expected to decrypt, embed and then re-encrypt the stream, which increased transmission latency and cost of the router. Since the intermediaries had access to the partially fingerprinted but unencrypted copies, this posed a security threat.

In Watercasting, d watermarked copies, a slightly larger number than the multicast tree depth, were multicast and each active router (AR) dropped a subset of copies arriving at its node. The unique 'drop pattern' became the fingerprint signature. Apart from the large logging requirements to capture the state of the drop pattern for a specific multicast session, the other issues were: 1) Increased communication overhead to relay drop decisions; 2) Effects of router malfunction and packet loss on fingerprint retrieval and latency.

2.3 Receiver Based Fingerprinting

In this framework the fingerprint casting process is shifted to the receivers from the network. Higher router costs, storage requirements and increased end-to-end latency offset the advantages of distributed watermarking schemes. Shifting the process to the receivers obviously raises questions pertaining to trust but is a less expensive and bandwidth efficient solution to the multicast fingerprinting problem.

1) Architecture of Parviainen et al.: Parviainen et al. [15], proposed an interesting and bandwidth scalable receiver fingerprinting scheme by creating two encrypted and watermarked copies of the same multimedia stream which was resilient to small scale collusion. A big advantage of this approach is that the embedding is done at the source and hence fingerprinting and encryption are treated as orthogonal processes, which makes it a good candidate for secure multicast distribution. The receivers are expected to download twice the amount of video information to cast a fingerprint, which makes it very difficult for users with less memory and bandwidth. The other challenge in this architecture is the key management. Since, every user must have a different set of stream decryption keys, every join/leave operation within the multicast group having N members would result in the refreshing of $N-1$ re-keying messages.

2) Chameleon: The ingenious chameleon cipher proposed by Ross Anderson [1] provided a computationally efficient way of performing fingerprinting and decryption by generating slightly different decryption key streams. However the stream cipher operated on raw PCM coded audio samples and the fingerprint was embedded in the least significant bits (LSBs), which made the embedding process less robust. The consequence was a large key size and susceptibility to small scale collusion attacks.

2.4 Joint Fingerprinting and Decryption in the Compressed Domain

The absence of any design rules which governed the generation of decryption keys in the chameleon cipher, the prospect of designing such joint fingerprinting schemes which operated on compressed data and the scope for identifying the challenges in this new framework provided us with the necessary motivation for exploring this new research problem, a general formulation of which is given in [11].

Due to the fusion of the two processes at the receiver, the encryption must be performed in the same domain as the watermark embedding, i.e. spatial or transform domain. A good partial encryption scheme minimizes the number of components which are encrypted and so preprocessing the stream to extract the smallest set of perceptually significant components is a necessity. On the other hand it is well known that watermarks are more robust when embedded in the perceptually significant components. This common goal simplifies the choice of selective encryption scheme, which can then be adapted to incorporate fingerprinting. Because of this integration, the only medium through which the fingerprint can be embedded at the receiver is via the decryption keys. To prevent the decryption/fingerprinting process from introducing any perceptible artifacts, perceptual models must be integrated into the decryption key design process. Additional features such as collusion resistance and error correction coding can be incorporated in the preprocessing of the decryption keys to improve error resiliency of the fingerprint.

3 JFD Principle and Tradeoffs

Here, the source extracts the perceptually relevant component X and selectively encrypts it using a key or set of keys K_E. Based on this parent key set, different decryption key sets $K_R = \{K_{D1}, K_{D2}, K_{D3}, \ldots, K_{Dn}\}$ are derived from the encryption key based on certain requirements out of which the most important is perceptual quality. The complete process is presented in Figure 2.

These keys K_R could be visualized as descendants of the encryption key set K_E as shown in Figure 3(a). The common trait is responsible for the restoration of the perceptual quality of the decrypted image while the difference is necessary for embedding a unique fingerprint. Alternatively, another way to visualize the decryption keys is as elements of overlapping subspaces $V(1), V(2), \ldots V(n) \subset V_E$ and $V(i) \cap V(j) \neq \phi$, all of which are contained in vector space V_E from which K_E is obtained. This is shown in Figure 3(b). Each subspace $V(i)$, has a unique basis $B(i)$ which can be used to reconstruct K_{Di} but not K_E. This error translates into a fingerprint upon decryption.

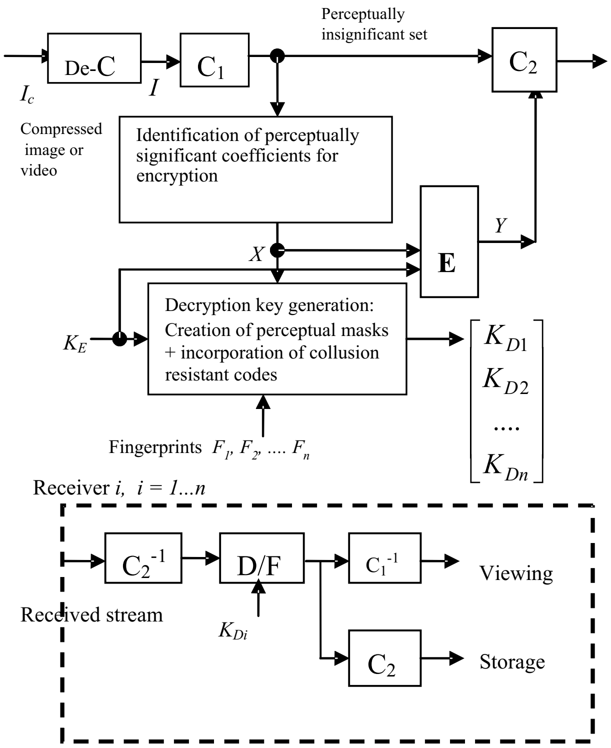

De-C
C1
C2
Perceptually insignificant set
I_c
I
Compressed image or video
Identification of perceptually significant coefficients for encryption
E
Y
X
K_E
Decryption key generation: Creation of perceptual masks + incorporation of collusion resistant codes
K_{D1}
K_{D2}
....
K_{Dn}
Fingerprints $F_1, F_2, ... F_n$
Receiver i, i = 1...n
C_2^{-1}
D/F
C_1^{-1}
Viewing
Received stream
K_{Di}
C2
Storage

Figure 2: Joint fingerprinting and decryption (JFD)

The subspaces $\{V(i)\}$ could be further subdivided in scenarios where secret sharing and fingerprinting are to be effected jointly. Presented below are some of the fundamental design requirements. The first two are critical for imperceptibility and robustness of the embedded fingerprint. The third constraint regarding component selection is important from the point of view of both fingerprinting and encryption and in some way dictates the manner in which the encryption operation must be carried out.

3.1 Perceptual Quality

A very important requirement in the joint design of the encryption and decryption keys is the condition of imperceptibility associated with the decrypted/fingerprinted image and that of obscurity associated with the encrypted image. Given a similarity metric $Simi$, thresholds T_{K1}, T_{K2} and an arbitrary decryption key K_D, the distortion constraints can be expressed as,

$$D_p(X_D, X) < \delta_{p1} \text{ for } Simi[K_E, K_D] > T_{k1} \quad (1)$$
$$D_p(X_D, X) > \delta_{p2} \text{ for } Simi[K_E, K_D] < T_{k2} \quad (2)$$

where, the perceptual distance metric $D_p()$ is application dependent with examples being the commonly used L^1, L^2 norms and PSNR and $X_D = D[Y, K_D]$ is the decrypted copy. The first constraint is necessary to ensure users with valid decryption keys, are able to decrypt the content properly without degrading the perceptual quality when a fingerprint is being embedded. If a particular decryption key K_D, is spurious i.e. has very few traits in common with K_E then decryption should result in a degradation of the image/video quality which is depicted in Condition (2). δ_{p1} is a global masking threshold for all the n fingerprints and δ_{p2} is the minimum perceptual distance necessary to irritate the viewer after encryption or if the wrong key is used for decrypting the stream.

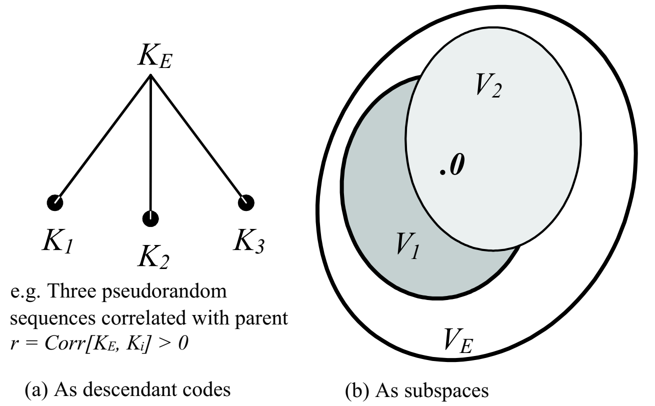

K_E
K_1
K_2
K_3
e.g. Three pseudorandom sequences correlated with parent
r = Corr[K_E, K_i] > 0
V_2
.0
V_1
V_E
(a) As descendant codes
(b) As subspaces

Figure 3: Few ways of visualizing the decryption key construction

3.2 Collusion Attacks and Collusion Resistance

In this framework, since decryption keys serve as fingerprint carriers, we perceive two types of collusion, one of which could happen in the key domain if the receiver has direct access to the decryption keys. Coming back to the significance of the thresholds T_{K1} and T_{K2}, we may consider two scenarios: (i) The decryption key K_D is some function of a subset of valid decryption keys (e.g. key collusion attack); (ii) The decryption key has no resemblance to the parent encryption key (i.e. is spurious or randomly generated). In Case (i), if the number of colluders is relatively small, the pirates will succeed in decrypting the copy with a distortion somewhere between δ_{p1} and δ_{p2} or even less than δ_{p1}. Hence, it is important to introduce collusion resistant coding methods to deter such key collusions. If K_D is randomly generated, then the distortion will be greater than δ_{p2} with a high probability. The two collusion attacks can be represented as,

1) *Type-A: Key collusion*

$$\hat{K}_D = \Phi[K_{p(1)}, K_{p(2)}, \ldots, K_{p(t)}]$$

where Φ can be any arbitrary linear or non-linear function such as 'XOR', 'majority bit vote', result of a 'random key mix' etc and the decryption keys $\{K_{p(1)}, K_{p(2)}, \ldots, K_{p(t)}\} \subset K_R$.

2) *Type-B: Linear collusion of decrypted copies*

$$\hat{X}_C = \frac{[X_{p(1)} + X_{p(2)} + \ldots + X_{p(t)}]}{t}.$$

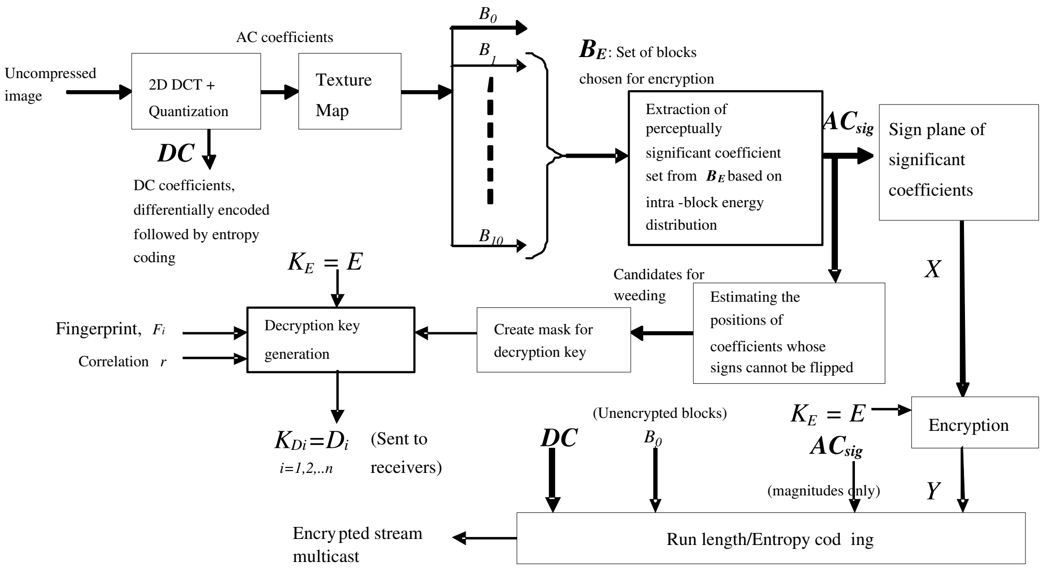

Uncompressed image
2D DCT + Quantization
AC coefficients
Texture Map
B_0
B_1
B_{10}
B_E: Set of blocks chosen for encryption
Extraction of perceptually significant coefficient set from B_E based on intra -block energy distribution
AC_{sig}
Sign plane of significant coefficients
DC
DC coefficients, differentially encoded followed by entropy coding
$K_E = E$
X
Candidates for weeding
Estimating the positions of coefficients whose signs cannot be flipped
Create mask for decryption key
Decryption key generation
Fingerprint, F_i
Correlation r
$K_{Di}=D_i$
$i=1,2,..n$
(Sent to receivers)
(Unencrypted blocks)
DC
B_0
$K_E = E$
AC_{sig}
(magnitudes only)
Encryption
Y
Encry pted stream multicast
Run length/Entropy cod ing

Figure 4: Joint fingerprinting based on sign bit encryption

Both type A and B attacks can be deterred by integrating collusion resistance into the decryption key designs.

3.3 Criteria for Selecting Components for Partial Encryption

If this choice has to be made in the absence of fingerprinting then options abound. The general criteria, which are used for selecting the components, are based on perceptual significance and resistance to error concealment attacks. However, the integration of fingerprinting restricts the number of possibilities such that Conditions (1) and (2) are met and a fairly robust fingerprint is embedded.

3.4 Inseparability

The decryption keys $K_{Di} = g[K_E, F_i]$ for $i = 1 \dots n$, are created by fusion of the encryption key with fingerprint information F_i (Figure 2). In cases where the entropy of K_E is much larger than the entropy of the embedded fingerprint the problem becomes equivalent to that of watermark embedding and the watermark is secured primarily because the search space is large but is vulnerable to key collusion attacks (which is one of the many ways of separating the two components). On the other hand, when the entropies become comparable, key collusion is likely to be less effective for the same number of colluders. The impact of the latter on the perceptibility of the fingerprint or the softening of the encryption process itself is another issue which is worth exploring especially in connection with the thresholds T_{K1} and T_{K2} mentioned in Section 3.1.

4 JFD Based on Sign Bit Encryption

The sign bit plane of perceptually significant AC coefficients is chosen for encryption and embedding. The following steps (shown in Figure 4) are executed after subjecting a raw image to block DCT and scalar quantization as specified in the JPEG still image compression standard.

4.1 Texture Map Creation

The objective here is to identify blocks, which have higher visual information (edges, texture, contrast variations etc). Due to frequency and contrast masking effects of the human eye these blocks also have a relatively high embedding capacity. For a given 8×8 block B_k where $k = 1, 2, \dots 1024$ for a 256×256 image, computation of the following parameters is necessary for the classification process,

$$\left.\begin{aligned}
&Tex(B_k) = \frac{E_{avg}(B_k)}{[E_{avg[over\ all\ blks]}]}\\
&Tex_{norm}(B_k) = \frac{Tex(B_k)}{[\max_{all\ blocks} Tex(B_k)]}\\
&Level_{tex}(B_k) = \lfloor \frac{Tex_{norm}(B_k)}{a} \rfloor \in \{0, 1, 2, \dots, \lfloor 1/a \rfloor\}\\
&E_{avg}(B_k) = \frac{1}{63}\sqrt{\sum_{i-1}^{63}[AC_i(B_k)]^2}
\end{aligned}\right\}$$

where AC_i represents the set of quantized AC coefficients. In our simulations we have set $a = 0.1$ which gives us 11 texture levels [0-10]. It has been a general observation that blocks containing sharp edges fall in levels $4 - 10$. The energy distribution in these blocks is usually concentrated within the first ten AC coefficients and is thus responsible for any ringing artifacts if any of these coefficients are disturbed because of fingerprinting. The energy distribution in blocks with high texture (levels 3-5) is usually shifted to the higher frequencies and these blocks have a relatively higher embedding capacity. The remaining blocks (levels 1-2) carry very little useful information and hence need not be encrypted.

4.2 Creation and Compression of the Encryption Map[B_E]

The binary encryption map is a two level re-quantization of the texture map based on the level-threshold specified by the source. Typically this threshold is chosen as ≥ 2 so that all the critical features are obscured. This map must be available at the receiver for synchronizing the decryption process and so must be compressed to reduce the communication overhead. Since, texture and edge bearing blocks in natural images exhibit a strong spatial correlation these small binary maps are compressible using standard methods such as run-length coding, minterm reduction, lossless binary wavelet transforms, parameterized morphological operators etc. The compressed encryption map may be piggybacked to the transmitted stream or embedded in the components which are not encrypted, the former preferred as perfect recovery of the map is important.

4.3 Extraction of Significant AC Coefficients

Having picked the blocks for encryption, the next obvious step is to select only those AC coefficients where the block energy is concentrated. Since most images tend to have a low pass characteristic, the cumulative energy distribution is computed for each block in B_E. The smallest coefficient number that corresponds to $85 - 90\%$ of the block energy is set as the block threshold w_k for $k = 1, 2, \dots h$, where h is the number of blocks chosen for encryption. The first w_k coefficients in each block in B_E can be extracted but the problem is that the $w_k s$ must also be sent to the receiver which adds to the transmission overhead. To reduce the overhead, the first w coefficients from these blocks are transmitted where w is the average of all the thresholds $\{w_1, w_2, \dots w_h\}$. The sign bit plane corresponding to this extracted coefficient set is represented as X.

4.4 Encryption and Decryption Key Generation

The encryption process is a modulo-2 product of X with the binary encryption matrix E which comprises of a set of independent and identically distributed random variables. The encrypted sign bit plane is given by,

$$Y = X \bullet E.$$

If the decryption matrix $D_i = E$ then no fingerprint will be embedded as the decrypted matrix will be $X_D = D_i \bullet Y = X$. However, if the decryption key D_i is made to differ from E in selective bit positions $\{f_1, f_2, \dots f_L\}$, which represents the fingerprint F_i, then we get a partially decrypted copy $X_i = D_i \bullet Y$. A large L implies greater robustness but a lower perceptual quality. The number of bit positions in which E and D_i differ is a function of the correlation r between the encryption and decryption sequences, which in turn affects the perceptual quality. The correlation $r = r_t$ is chosen so that the fingerprinted copy has a perceptual quality $> Q2$ as shown in Figure 5.The case $r = 0$ is equivalent to encryption. By encrypting texture levels > 1 we can ensure in most cases that $Q1$ corresponds to PSNR value of about $20dB$. The gap $Q2 - Q1$ increases with r and has been observed to be relatively independent of the underlying image characteristic. We choose $r = r_t$ so that the fingerprint is just imperceptible to maximize the embedded payload.

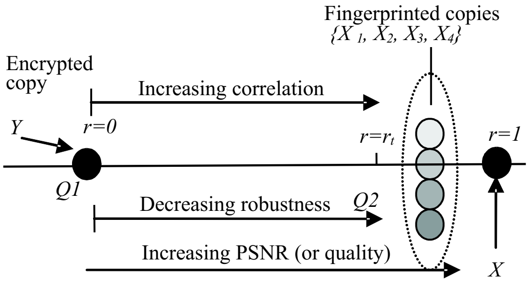


Figure 5: Effect of correlation on quality of the fingerprinted copy

There are however two major issues which influence the preprocessing of the decryption keys: 1) Reduction of block artifacts introduced by fingerprinting, 2) Collusion resistance. The first problem arises because of the implicit assumption that all the coefficients have virtually the same perceptual significance, which is reflected from choice of parameter r, which is used to control the perceptual quality of the fingerprinted copy. The first $10 - 12$ AC coefficients dominant in edge carrying blocks have very low JND thresholds. Any forced sign changes in this set will result in ringing effects or artifacts. Hence the goal behind the key massage is to identify these coefficients and weed them out of the fingerprinting domain. The challenge is in estimating the positions of these components, which are responsible for the artifacts with minimal computation and memory usage. Aggressive weeding reduces the fingerprint payload considerably but is recommended for very sensitive documents such as medical images.

4.5 Collusion Resistance

The fingerprint F_i comprises of bit positions $\{f_1, f_2, \dots f_L\}$ at which the encryption and decryption keys do not match. If we define the position matrix

Table 1: Code design for collusion resistance

	Marks or basis vectors					
Users$(n = 4)$	1	2	3	4	5	6
A	X	X	X			
B	X			X	X	
C		X		X		X
D			X		X	X

$P = \{1, 2, 3, \ldots hw\}$ then $F_i \subset P$. A sufficient condition for orthogonal embedding is $F_i \cap F_j = \phi, \forall i \neq j$. However orthogonal embedding is highly susceptible to collusion attacks. Here less than five users can collude their copies to erase the fingerprint. Hence there is a need for a mechanism for tracking subsets of colluders. We present a simple coding scheme for a small user space $n = 4$ in which all possible two-collusions can be detected.

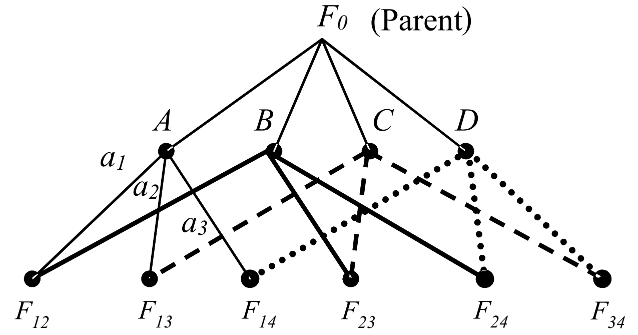


Figure 6: Collusion resistance

This simple code design method follows the graph depicted in Figure 6. In the first level we have the user-fingerprints. All 2-collusion possibilities with a particular fingerprint i, are treated as children $\{ch1(i), ch2(i), ch3(i)\}$ of i. Each 2-collusion, which is a child to two parents must exhibit some common trait seen in both the parents. For example, the child F_{24}, which is spawned by the fusion of parents u_2 and u_4, will exhibit a trait, which is common to both and will uniquely identify this collusion couplet. The four fingerprints $\{A, B, C, D\}$ are split as $A = \{a_1, a_2, a_3\}, B = \{b_1, b_2, b_3\}, C = \{c_1, c_2, c_3\}, D = \{d_1, d_2, d_3\}$. The inheritance constraint asserts that $a_1 = b_1 = 1; a_2 = c_1 = 2; a_3 = d_1 = 3; b_1 = a_1 = 1; b_2 = c_2 = 4; b_3 = d_2 = 5; c_1 = a_2 = 2; c_2 = b_2 = 4; c_3 = d_3 = 6; d_2 = b_3 = 5; d_1 = a_3 = 3; d_3 = c_3 = 6;$ The consequence is Table 1. So six basis vectors or marks $\{1, 2, 3, 4, 5, 6\}$ are required to generate this anti-collusion code to track users $\{u_1, u_2, u_3, u_4\}$ and all 2-collusion pairs $\{(u_1, u_2), (u_1, u_3), (u_1, u_4), (u_2, u_3), (u_2, u_4), (u_3, u_4)\}$.

5 Fingerprint Detection

Let I_O be the original un-watermarked image with the source, I_E the encrypted copy and I_{FU} represent the decrypted copy embedded with fingerprint F_U. Let I_R be the image retrieved by the source whose origin must be traced. An estimate of the fingerprint embedded in I_R can be obtained by comparing the sign maps of the original with that of I_R, i.e.,

$$\hat{F} = Position\_diff[X, X_R]$$

where, $Position\_diff(.)$ is a function which determines the set of positions at which host sign plane X differs from that of the retrieved sign plane X_R. With respect to the originally embedded fingerprint F_U, the retrieved fingerprint can be expressed as two disjoint sets: (i) set of positions in the domain of the original fingerprint F_U which have not been flipped because of post-processing; (ii) the positions outside the domain of F_U, which have been affected. This interference in the detection process can be modeled as noise, which is represented by the set $V \subset P$. Thus, we may express the retrieved fingerprint as a set difference between the originally embedded fingerprint F_U and the noise set V,

$$\hat{F} = F_U - V = [F_U \cap V^c] \cup [F_U^c \cap V].$$

Our main objective in the detection process is in identifying F_i which maximizes the likelihood, $P(\hat{F}/F_i)$. This function can be further simplified as,

$$\begin{aligned} P(\hat{F}/F_i) &= P[F_U - V/F_i] = \frac{P[(F_U - V) \cap F_i]}{P(F_i)} \\ &= \frac{P[F_i \cap \{[F_U \cap V^c] \cup [F_U^c \cap V]\}]}{P(F_i)} \\ &= \frac{P[F_i \cap (F_U \cap V^c)] + P[F_i \cap (F_U^c \cap V)]}{P(F_i)}. \quad (3) \end{aligned}$$

The last step results because sets $F_U \cap V^c$ and $F_U^c \cap V$ are disjoint. The set $\{P(F_i)\}$, represents the priory knowledge about the fingerprints which are likely to be embedded in the retrieved copy. If no such information is available then we may set $P(F_i) = 1/N_f$ and ignore that term for future analysis. The priory distribution $\{P(F_i)\}$ may be skewed based on the list of suspects. To simplify our analysis, we start by assuming that no such priory information is available to the detector. Equation (3) then simplifies to,

$$\begin{aligned} &{}^{Arg}_{F_i} \max[P(\hat{F}/F_i)] = \\ &\quad {}^{Arg}_{F_i} \max[P(F_i \cap (F_U \cap V^c)) + P(F_i \cap (F_U^c \cap V))]. \end{aligned}$$

Assuming orthogonal embedding we have $F_i \cap F_j = \phi$ for ij. Based on the orthogonal condition we have two different cases: (i) $F_i = F_U$ and (ii) $F_i \neq F_U$. So the sum simplifies to,

$$\begin{aligned} &P(F_i \cap (F_U \cap V^c)) + P(F_i \cap (F_U^c \cap V)) \\ &\quad = P(\phi) + P(F_i \cap V), \ \forall i \neq U \\ &\quad = P(F_i \cap V^c) + P(\phi), \ i = U. \end{aligned}$$

The term $P(\phi)$ is common and can be ignored in future analysis. Since the cases, $F_i = F_U$ and $F_i \neq F_U$, represent the cases where the correct fingerprint has been detected and a false positive is incurred respectively, we may rewrite these probability distributions for correct detection and false positive respectively as,

$$f(\hat{F}/Correct) = P(F_i \cap V^c) \quad (4)$$
$$f(\hat{F}/FP) = P(F_i \cap V). \quad (5)$$

Our next step is to identify the minimum threshold so that the likelihood ratio (LR), $f(\hat{F}/Correct)/f(\hat{F}/FP)$ is equal to one. Further increase in the threshold T will decrease the false positive rate but will increase the probability of a false negative. Hence selection of T is based on a tradeoff between what we may define as acceptable false positive and negative rates for this application. For high security applications such as joint access of sensitive records, high threshold is preferred for security reasons at the cost of increasing the false negative rate. On the other hand in applications where identifying as many suspects as possible is important, the threshold value can be low. The price paid here is a poor detection accuracy.

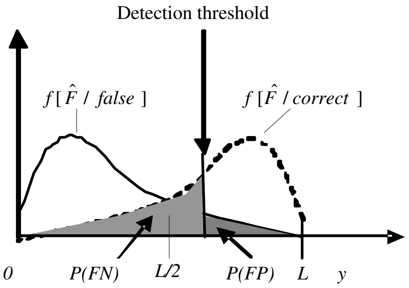


Figure 7: Showing the FP and FN probabilities

Having defined the performance requirements of our fingerprinting system, we now need a model for the noise or interference. The noise is the result of a cumulative effect of any form of post processing such as image enhancement, compression or collusion on the original fingerprinted copy X_{FU}. We may represent the noise as a $h\ x\ w$ matrix G which comprises of a set of independent and identically distributed (IID) binary random variables g_{ij}, such that $P(g_{ij} = -1) = \varepsilon$. The effect of noise can be approximated as,

$$X_R = X_{F_U} \bullet G.$$

The noise set V and the noise matrix G is equivalent. The set V comprises of all those positions in chosen sign bit plane which are likely to be affected. Thus V is derived from G as,

$$\begin{aligned} V &= Positions[(i-1)w + js.t.g_{ij} \\ &= -1,\ i = 1 \ldots h,\ j = 1 \ldots w]. \end{aligned}$$

Based on the error characteristics of the noise matrix (i.e.), a few or most of the elements in F_i may be affected. By assuming that the noise is IID, we impose the condition that the error characteristics are independent of the bit position. Following this we can now partition the outcomes of the event $F_i \cap V$ into $L + 1$ equivalence classes such that all sets within a particular equivalence class have the same probability of occurrence. However collectively, since the number of bit errors within the fingerprint domain is a function of ε, we anticipate that the probability that $F_i \cap V$ will fall in a particular equivalence class is likely to vary. The equivalence classes have been chosen on the basis of their cardinality, i.e. we may say,

$$F_i \cap V \in A_K \text{ if } \mid F_i \cap V \mid = K$$

where, A_K = Collection of sets which have K elements, $K = 0K.L$ and $A_0 = \phi$.. So with this reasoning, we represent $\mid F_i \cap V \mid$ by a random variable Z, which gives,

$$P[F_i \cap V \in A_K] = P[Z = K]$$

We will assume the distribution of Z to be of binomial type (n, p, q) with $n = L, p = \varepsilon$ and $q = 1 - \epsilon$. The probability $P[Z = k]$ represents the number of bit errors introduced in F_i.

$$P[F_i \cap V \in A_K] = P[Z = k] = \binom{L}{K}\varepsilon^k(1-\varepsilon)^{L-k}. \quad (6)$$

In a similar vein, in the event of a correct detection, we can write the distribution for $P(F_i \cap V^c)$ as binomial $(L, 1 - \varepsilon, \varepsilon)$,

$$P[F_i \cap V^c \in A_K] = P[W = k] = \binom{L}{K}(1-\varepsilon)^k\varepsilon^{L-k}. \quad (7)$$

Using Equations (4, 5, 6) and (7) we may plot the distributions as shown in Figure 7. for $\varepsilon << 0.5.X, X_U$ and X_R are the $h\ x\ w$ sign matrices extracted from the original image I_O, fingerprinted copy I_{FU} and retrieved copy I_R respectively. The corresponding fingerprints retrieved from these matrices are given by,

$$\begin{aligned} F_U &= Pos\_signdiff\lfloor X_{F_U}, X \rfloor \\ \hat{F} &= Pos\_signdiff[X, X_R] \end{aligned}$$

From the above result, we may compute empirically,

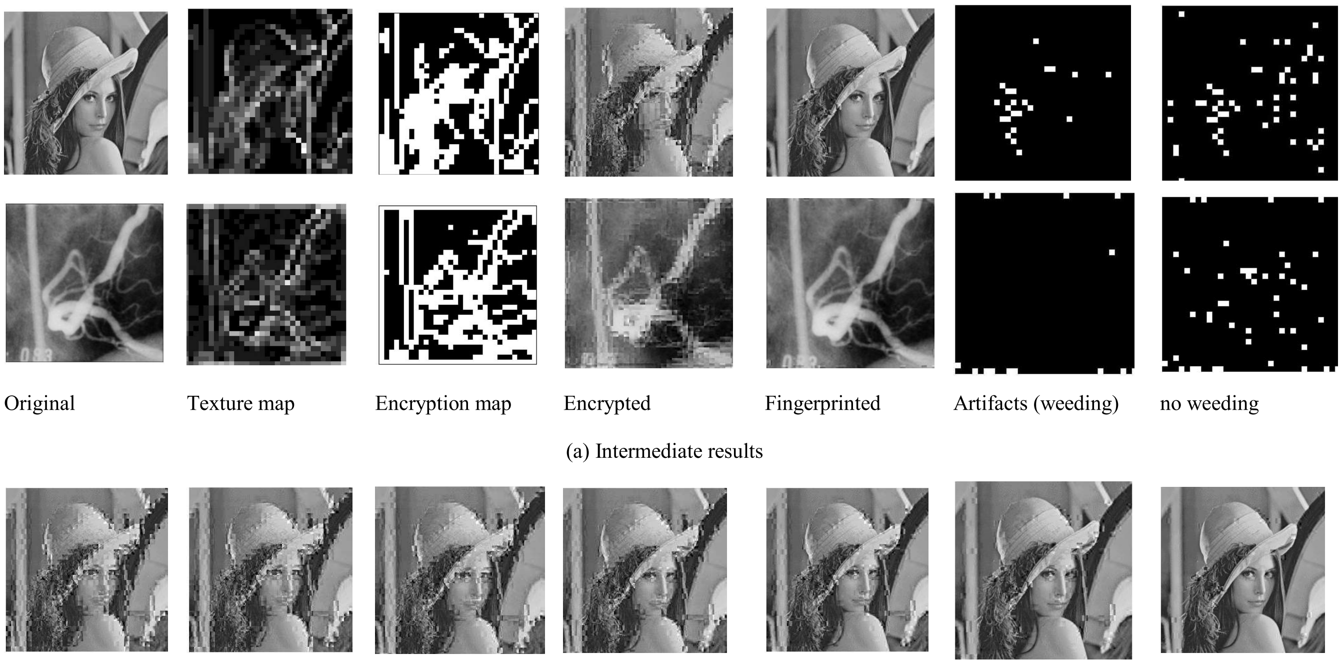
$$y_i = \mid \hat{F} \cap F_i \mid \text{ for } i = 1 \ldots N_f.$$

The observation y_i can be further simplified as,

$$\begin{aligned} y_i &= \mid \hat{F} \cap F_i \mid = \mid F_i \cap (F_U - V) \mid \\ &= \mid F_i \cap [(F_U \cap V^c) \cup (F_U^c \cap V)] \mid \\ &= \mid (F_i \cap F_U \cap V^c) \cup (F_i \cap F_U^c \cap V) \mid \quad (8) \end{aligned}$$

For the cases where $F_i = F_U$ and $F_i \neq F_U$ (Equation 8) reduces to,

$$\begin{aligned} y_i &= \mid F_i \cap V^c \mid = W\ i = U \\ &= \mid F_i \cap V \mid = Z\ \forall i \neq U. \end{aligned}$$



(a) Intermediate results

(b) Effect of increasing correlation, 0.05-0.95 in steps of 0.15 on image quality [PSNR increases from *18.11 dB to 31.11dB*]

Figure 8: Results for the sign bit scheme

The observation may belong to the distribution f_W or f_Z based on whether $F_i = F_U$ or $F_i \neq F_U$. To ensure $f_W(y_i) \geq f_Z(y_i)$ we should set the detector threshold as $T \geq L/2$ which becomes clear from Figure 7. In our simulations, we have set $T = 0.55L$. For a given detector threshold T and fingerprint length L, we can write the false positive and negative probabilities as,

$$\begin{aligned} P[FN] &= \sum_{k=0}^{T} \binom{L}{K} (1-\varepsilon)^k \varepsilon^{L-k} \\ P[FP] &= \sum_{k=T+1}^{L} \binom{L}{K} \varepsilon^k (1-\varepsilon)^{L-k}. \end{aligned}$$

In the images used, the fingerprint length L was found to be 834 and 397 respectively for Lena and Angiogram images respectively as per Table II. Thus for values of which are not very small, we may approximate the binomial distribution as a Gaussian distribution with parameters $\sigma = \sqrt{Lpq}$and $\mu = Lp$ based on DeMoivre-Laplace theorem, where $p = 1 - \varepsilon$ or ε depending on whether the distribution for correct detection or false positive is used. When the sign bit error approaches 0.5 we may approximate the FN and FP probabilities respectively as,

$$\begin{aligned} P(FN) &\approx G[\frac{T - L(1-\varepsilon)}{\sqrt{L\varepsilon(1-\varepsilon)}}] - G[\frac{-L(1-\varepsilon)}{\sqrt{L\varepsilon(1-\varepsilon)}}] \\ &\approx G[\frac{T - L\varepsilon}{\sqrt{L\varepsilon(1-\varepsilon)}}] \\ P(FP) &\approx 1 - G[\frac{T - L\varepsilon}{\sqrt{L\varepsilon(1-\varepsilon)}}] \end{aligned}$$

where $G(x)$ is the cumulative distribution function. So for values of ε around 0.5 a tradeoff between FP and FN rate is achieved by varying the threshold T from 40 to 60% of the fingerprint length L. For $\varepsilon << 0.5$, the question of a tradeoff between FP and FN does not arise.

A simple way to analyze the effect of anti-collusion codes on the FP and FN rates is by imposing the condition $F_i \cap F_j = A_{ij} \neq \phi$ where the set A_{ij} has three disjoint components, two of which uniquely identify users i and j and the third represents a portion common only to the pair $[i, j]$. The fingerprint payload is distributed amongst these three components based on a tradeoff between resolution and maximizing the number of suspects detected. With further nesting this methodology can be extended to detect an arbitrary number $t < N_f$ colluders.

Table 2: Important parameters

	Lena	Angiogram
PSNR(encrypted)	17.88dB	20.9604 dB
PSNR(fingerprinted)	29.3108dB	34.1022dB
	(r = 0.88)	(r=0.85)
Artifacts with weeding	20	14
Encryption/decryption matrix size [h,w]	[662,21]	[757,7]
Fingerprint lengths (L)	834	397

Table 3: Resistance to recompression [LENA]

Compression	$y_i = \frac{\lvert F_i \cap \hat{F}\rvert}{\lvert F_i\rvert}$, $i = A, B, C, D$			
$[Q, PSNR]$	**A**	**B**	**C**	**D**
90, 28.695dB	**0.8230***	0.1050	0.1099	0.0916
70, 27.795dB	**0.7106**	0.2381	0.1978	0.1893
50, 27.312dB	**0.6422**	0.2882	0.2674	0.2308
40, 27.066dB	**0.6361**	0.3077	0.2784	0.2686
30, 26.715dB	**0.5934**	0.5934	0.3077	0.3016
20, 26.192dB	**0.5653**	0.3712	0.3297	0.3297
10, 25.151dB	0.5140	0.4310	0.3858	0.3687
5, 23.408dB	0.5031	0.4640	0.4310	0.4371

*Detected fingerprints marked in bold font.
Size of the sign plane [662, 21], r =0.88, Threshold = 55% and weeding step used. Artifact count = 20; Detection failed for $Q = \{10, 5\}$.

6 Simulations and Analysis

6.1 Results

PSNR values of the encrypted and fingerprinted images are presented in Table 2. Two images with slightly different characteristics have been chosen for simulation. For the simulations we have chosen a small user space of four $\{A, B, C, D\}$. Tables 3 and 4 indicate the effect of different degrees of compression (represented by quality factor Q), on the detection process. Fingerprint of user A is embedded when the images are decrypted. Figure 8(a), shows the encryption and fingerprinting results for two different images while Figure 8 (b) demonstrates the effect of increasing correlation r on the PSNR of the decrypted image.

Collusion resistance has been incorporated based on the coding methodology discussed in Section 4.5, in which all two-collusions in a four user space are detectable. This is implemented by first creating six disjoint fingerprint sets or marks $\{1, 2, 3, 4, 5, 6\}$ which correspond to six basis vectors. Each embedded codeword has three symbols or marks selected as per Table I. The collusion resistant codes have been designed so that common traits (or marks) are preserved and uncommon traits are erased with a high probability when any two users collude. For instance when user A with $\{1, 2, 3\}$ and user C with $\{2, 4, 6\}$ collude their copies we anticipate that $\{2\}$ will be the only mark detected in the retrieved document which will uniquely identify the pair $[A, C]$. So both fragility and robustness are important in the design of effective anti-collusion codes - erasure of uncommon marks is just as important as preservation of common ones to narrow down the list of suspects. In Table 5, the test results of compression, collusion and combinations of the two on the detection of the fingerprint are presented. Effects of both key and linear collusion attacks are shown.

Table 4: Resistance to recompression [MEDICAL IMAGE]

Compression	$y_i = \frac{\lvert F_i \cap \hat{F}\rvert}{\lvert F_i\rvert}$, $i = A, B, C, D$			
$[Q, PSNR]$	**A**	**b**	**C**	**D**
90, 35.2388dB	**0.6808**	0.0647	0.0692	0.0580
70, 34.4766dB	**0.6272**	0.1049	0.1205	0.1183
50, 33.7800dB	**0.5670**	0.1429	0.1808	0.1607
40, 33.3338dB	0.5402	0.1696	0.1897	0.2054
30, 32.6803dB	0.5156	0.2098	0.2299	0.2455
20, 31.5076dB	0.4866	0.2612	0.2790	0.2701
10, 29.2694dB	0.4397	0.3482	0.3616	0.3482
5,25.9808dB	0.4665	0.4241	0.4464	0.4174

Size of the sign plane [757, 7], r = 0.85, Threshold = 55% and weeding step used. Artifact count = 14; Detection failed for $Q < 50$.

6.2 Secrecy and Key Size

The encryption and fingerprinting processes are modulo-2 matrix products $Y = X \bullet E$ and $X_D = Y \bullet D$. An eavesdropper has access to the encrypted image I_E, but does not know the encryption map. However, he may succeed in estimating it fairly accurately since the texture map is created based on the magnitudes of the AC coefficients and the coefficients are not shuffled across blocks. So assuming now that he has the map, it may not be unreasonable to assume that he will then extract the sign matrix Y. So the security of this scheme comes down to the estimation of E or X from Y.

If all entries of E are binary IID random variables with a distribution $f = \{1/2, 1/2\}$, then $H[E]$ is maximum and equal to hw bits. This guarantees information theoretic secrecy since $H[X]_{max} = hw$ over all possible choices for a binary random process X. It is thus theoretically impossible for the attacker to retrieve X from Y. In general, to preserve theoretic secrecy [18], the entropy rate of the key must be greater than or equal to the entropy rate of X, i.e.,

$$\lim_{n\to\infty} \frac{H[X_1, X_2, \ldots, X_n]}{n} \leq \lim_{n\to\infty} \frac{H[E_1, E_2, \ldots, E_n]}{n}$$

Hence if the evolution of the discrete process X can be tracked, an encryption key much smaller than hw bits will

Table 5: Combined effect of collusion and compression

	Detection of embedded marks M_i: $y_i = Proj(M_i, \hat{F}) = \frac{\lvert M_i \cap \hat{F}\rvert}{\lvert M_i\rvert}$, $i = 1, 2, 3, 4, 5, 6$					
	1	2	3	4	5	6
$C(X_A, 40)$	**0.7441**	**0.7475**	**0.7071**	0.2593	0.2862	0.2795
	Detected = [1,2,3] and user A identified correctly					
$Col[C(X_A, 40), C(X_C, 70)]$	0.4680	**0.8081**	0.4343	0.5084	0.2054	0.5387
	Detected = [2], Possible suspects are {A,C}					
$Col[C(X_A, 30), C(X_B, 50)]$	**0.7643**	0.4916	0.4512	0.5017	0.4747	0.2424
	Detected = [1], Possible suspects are {A,B}					
$C(X_A \rightarrow KCol[X_A, X_B, X_D], 90)$	**0.8889**	0.1347	0.5387	0.4040	**0.8653**	0.1246
	Detected = [1,5], Possible suspects are {A,B,D} or {B}					
$C(X_B \rightarrow KCol[X_B, X_C, X_D], 90)$	0.2795	0.2593	0.2660	**0.7239**	**0.7677**	**0.7677**
	Detected = [4,5,6], Possible suspects are {B,C,D} or {B} or {C}. Note here that if just B and C had colluded then only [4] would have been detected, hence the pair {B,C} has not been included in the list of suspects					

$Col(X, Y, Z) \Rightarrow$ linear collusion of fingerprinted copies X, Y and Z
$KCol(X, Y, Z) \Rightarrow$ collusion of decryption keys through bit voting
$C(X, Q) \Rightarrow$ JPEG compression to quality factor Q.
Fingerprints for users A, B, C, D are: $\{1, 2, 3\}, \{1, 4, 5\}, \{2, 4, 6\}$ and $\{3, 5, 6\}$ respectively.

be required protect the stream. But this may not be easy to compute because of the non-stationary characteristic of X. For the JFD scheme based on sign bit embedding, the side information required for a 256 × 256 image is $hw/256^2bpp$ plus the information needed to encode the encryption map. The condition for perfect secrecy can be relaxed to reduce the size of the encryption key.

7 Conclusions

By integrating perceptual models indirectly through a weeding process and collusion resistant coding methodologies in the design of the decryption keys, we have shown that it is possible to embed a robust yet relatively imperceptible fingerprint at the receiver. The encryption process is softened due to the incorporation of perceptual constraints which is depicted in Equations (1) and (2). A deeper understanding of this softening problem will provide us with some useful insights into the mechanism of joint fingerprinting for establishing some performance limits for JFD schemes.

Although we have not tested the proposed JFD algorithm based on sign bit embedding for all Stirmark attacks, we have shown that it is robust to recompression, small scale collusion and a combination of two. We are currently exploring decryption key design scenarios based on other transform domain compression/encryption methodologies such as encryption of wavelet packet structures. In this paper we discussed one potential application for JFD as multicast content protection. However, it is possible to extend the joint fingerprinting idea to applications such as access control and protection of sensitive records. We perceive that this framework can be used to merge secret sharing methods for joint access control and content fingerprinting.

Acknowledgements

This research work was supported in part by Bell University Laboratories and by CITO.

References

[1] R. J. Anderson and C. Manifavas, "Chameleon – A new kind of stream cipher", in *Proceedings Fourth Workshop on Fast Software Encryption*, pp. 107-113, Jan 1997.

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

[3] D. Boneh and J. Shaw, "Collusion secure fingerprinting for digital data," *IEEE Transaction on Information Theory*, vol. 44, pp. 1897-1905, Sep 1998.

[4] J. Brassil, S. Low, N. Maxemchuk, and L. O'Gorman, "Electronic marking and identification techniques to discourage document copying," in *Proceedings IEEE INFOCOM '94*, vol. 3, pp. 1278-1287, Toronto, Canada, June 1994.

[5] I. Brown, C. Perkins, and J. Crowcroft., "Watercasting: Distributed watermarking of multicast multime-

dia", in *First International Workshop on Networked Group Communication (NGC99)*, pp. 286-300, 1999.

[6] B. Chor, A. Fiat, and M. Naor, "Tracing traitors," in *Proceedings 14th International Cryptology Conference CRYPTO'94*, vol. 839, pp. 257-270, California, USA, Aug. 1994.

[7] C. S. Collberg and C. Thomborson, "Watermarking, tamper-proofing and obfuscation - Tools for software protection," *IEEE Transaction on Software Engineering*, vol. 28, no. 8, pp. 735-746, Aug. 2002.

[8] J. Haitsma and T. Kalker, "A highly robust audio fingerprinting system," in *Proceedings of ISMIR 2002*, pp. 107-115, Paris, France, Oct. 2002.

[9] N. Jayant, J. Johnston, and R. Safranek, "Signal compression based on model of human perception," in *Proceedings of the IEEE*, vol. 81, pp. 1385-1422, Oct. 1993.

[10] P. Judge and M. Ammar, "WHIM: Watermarking multicast video with a hierarchy of intermediaries", *Journal of Computer Networks*, vol. 39, no. 6, pp. 699-712, 2002.

[11] D. Kundur and K. Karthik, "Video fingerprinting and encryption principles for digital rights management," in *Proceedings of IEEE*, vol. 92, no. 6, pp. 918-932, Jun 2004.

[12] J. Lach and W. H. M. Smith, "Fingerprinting digital circuits on programmable hardware," in *Proceedings Second Int. Workshop on Information hiding 1998*, vol. 1525, pp. 16-31.

[13] L. Tang, "Methods for encrypting and decrypting MPEG video data efficiently ", in *Proceedings of the ACM Multimedia 1996*, pp. 219-229, Nov 1996.

[14] S. Lian and Z. Wang, "Comparison of several wavelet coefficient confusion methods applied in multimedia encryption," in *Int. Conf on Computer Networks and Mobile Computing*, pp. 372-376, Shanghai, China, Oct. 2003.

[15] R. Parviainen and R. Parnes, "Large scale distributed watermarking of multicast media through encryption", in *Proceedings Int. Conf on Communications and Multimedia Security 2001*, pp. 17-30, Deventer, Netherlands, May 2001.

[16] C. E. Shannon, "Communication theory of secrecy systems", *Bell System Technical Journal*, vol. 28-4, pp. 656–715, 1949.

[17] W. Trappe, M. Wu, Z. Wang, and K. J. R. Liu, "Anti-collusion fingerprinting for multimedia", in *IEEE Transaction on Signal Processing*, vol. 51, no. 4, pp 1069-1087, Apr2003.

[18] C. P. Wu and C. C. J. Kuo, "Efficient multimedia encryption via entropy codec design", in *Proceedings SPIE Security and Watermarking of Multimedia Content III*, vol. 4314, pp. 128-138, 2001.

Kannan Karthik (Student Member, IEEE), received his bachelors degree (B. E.) in electronics from Bombay University in 1998 and M.Eng degree in Power Electronics and Controls from Memorial University, St. Johns, Canada in 2001. He is currently working towards his Ph.D degree in Multimedia Security at the University of Toronto, Toronto, Canada.

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

Dimitrios Hatzinakos received the Diploma degree from the University of Thessaloniki, Greece, in 1983, the M.A.Sc degree from the University of Ottawa, Canada, in 1986 and the Ph.D. degree from Northeastern University, Boston, MA, in 1990, all in Electrical Engineering. In September 1990 he joined the Department of Electrical and Computer Engineering, University of Toronto, where now he holds the rank of Professor with tenure. Also, he served as Chair of the Communications Group of the Department during the period July 1999 to June 2004.

Since November 2004, he is the holder of the Bell Canada Chair in Mutimedia, at the University of Toronto. His research interests are in the areas of Multimedia Signal Processsing and Communications. He is author/coauthor of more than 150 papers in technical journals and conference proceedings and he has contributed to 8 books in his areas of interest. His experience includes consulting through Electrical Engineering Consociates Ltd. and contracts with United Signals and Systems Inc., Burns and Fry Ltd., Pipetronix Ltd., Defense Research Establishment Ottawa (DREO), Vaytek Inc., Nortel Networks, Vivosonic Inc. and CANAMET Inc.

He has served as an Associate Editor for the IEEE Transactions on Signal Processing from 1998 till 2002 and Guest Editor for the special issue of Signal Processing, Elsevier, on Signal Processing Technologies for Short Burst Wireless Communications which appeared in October 2000. He was a member of the IEEE Statistical Signal and Array Processing Technical Committee (SSAP) from 1992 till 1995 and Technical Program co-Chair of the 5th Workshop on Higher-Order Statistics in July 1997. He is a senior member of the IEEE and member of EURASIP, the Professional Engineers of Ontario (PEO), and the Technical Chamber of Greece.