Joint Fingerprinting and Decryption for Multimedia Content Tracing in Wireless Networks

Kannan Karthik[a] , Deepa Kundur[b], Dimitrios Hatzinakos[a]
[a]Dept. of Elect. and Computer Engg., University of Toronto, Toronto, Canada M5S3G4;
[b]Dept. of Elect. Engg., Texas A&M University, College Station, USA 77843-3128

ABSTRACT

Wireless Multimedia applications increasingly demand low power and bandwidth efficient techniques for content protection. Existing partial encryption methodologies juggle with the difficulties of trading decoding complexity for a higher bit rate, maintaining format compliance yet minimizing the number of components that need to be encrypted, to produce the desired distortion. Source based fingerprinting schemes for content tracking are neither bandwidth efficient nor computationally efficient when coupled with encryption. Thus inspired by the chameleon cipher we present a new algorithm for fingerprinting JPEG compressed images and a simple analytical model based on joint fingerprinting and decryption, as a computationally and bandwidth efficient solution for content protection and tracking.

Keywords: Fingerprinting, partial encryption, watermarking, chameleon cipher, JPEG compression

1. INTRODUCTION

Limited power and low bandwidth are the primary constraints in a wireless network especially when distributing large video, image and audio files in a secure way. Hence encryption schemes must be designed to encrypt only a small and the most sensitive subset of the bit-stream.

In a variety of commercial and non-commercial applications, it is as important to encrypt data as it is to track the content so that in the event of any tampering or malpractice such as unwarranted redistribution of 'sensitive' medical records, the content can be traced back to the user who received it first. Multimedia data tracking is done through a process called digital fingerprinting in which a robust and imperceptible watermark is embedded in the multimedia object to identify the user. The fingerprint is expected to robust to basic signal processing attacks such as geometric transformations like scaling and rotation[1], linear filtering and de-synchronization attacks[2], and also incidental distortions due to recompression, trans-coding etc. Some of the fingerprinting solutions attempt to improve watermark robustness to these basic attacks by increasing watermark diversity using spread spectrum techniques[3-4]and by preprocessing watermarks in the bit domain with channel codes such as BCH and Reedsolomon[5-6]. But the absence of an appropriate attack model due to the sheer breadth of attacks makes these approaches suboptimal. These methods also do not address the problem pertaining to the large scale collusion of fingerprinted copies.

To counter the powerful collusion attack, a variety of interesting schemes were proposed. Owing to the large computational complexity associated with the detection of orthogonal fingerprints coded modulation schemes were developed to reduce the number of correlations. Some of them were based on anti-collusion and frameproof codes[7-8], but these techniques were not effective in practice since they made certain assumptions regarding the behavior of the fingerprint in the bit domain, when subjected to linear collusion attacks. It was observed that for large groups, orthogonal fingerprinting is not feasible since the false positive and negative rates become almost equal. Hence the notion of group based fingerprinting using correlated fingerprints was introduced to narrow the search to a subset of colluders[9]. Su et al.[10] proposed a method that made the watermark statistically invisible to the colluders by allowing the watermark statistics to evolve with the host video frames.

Unfortunately in multicast environments, these source based fingerprinting schemes do not scale well in terms of bandwidth and computational complexity especially when coupled with encryption. Attempts have been made to mitigate the computational complexity problem by distributing the watermark trace over a set of routers[11-12], but this raises issues pertaining to trust and also some fundamental QoS issues such as increased end to end latency, since the content must be decrypted at each intermediate point before embedding a portion of the trace.

The ingenious chameleon cipher proposed by Anderson et al.[13] for audio distribution provided a way to merge two seemingly orthogonal techniques of watermarking and encryption by distributing slightly different decryption streams to

Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense III, E. M. Carapezza, ed., Proc. of SPIE Vol. 5403 (SPIE, Bellingham, WA, 2004) · 0277-786X/04/$15 · doi: 10.1117/12.542777

different receivers for a given encryption stream. Since only the least significant bits of the PCM coded audio are modified by the key stream, erasing these bits will not affect the perceptual quality of the audio stream and so the fingerprint is not robust. The scheme also assumes the audio to be in the raw form rather than the compressed form, which makes the multicast transmission infeasible. This scheme is extremely vulnerable to collusion, and it was shown in Ref. 13 that five or more users can produce a plain text that cannot be traced by merely using a bit-vote to erase the watermark from the LSBs.

Hence the main problem in combining the two processes is the difficulty in embedding a robust and imperceptible watermark within the domain of encryption, by cleverly designing decryption keys. To reduce the computational time and decoding complexity in secure wireless and wireline networks, partial encryption schemes that encrypt sensitive portions of the media stream were proposed. We present an overview of some of the design challenges in such schemes in Section 2 and also how this may influence the fingerprinting process. In essence, what we observe is that to perform fingerprinting in the same domain as encryption, we must compromise on both encryption complexity and watermark robustness. Thus bandwidth efficiency and reduction in computational complexity comes at a price of a weakening of the encryption process and also a compromise on the payload and robustness of the fingerprint that is embedded at the receiver.

In this paper we capture the spirit of the chameleon and explore this new area of joint fingerprinting and decryption (JFD) with the objective of developing an architecture for creating audit trails. We present a simple analytical model to illustrate the tradeoffs in this environment and in Section 4 present an algorithm and some simulation results for joint fingerprinting and decryption of JPEG images.

2. PARTIAL ENCRYPTION AND FINGERPRINTING

The primary goal behind designing partial encryption (PE) schemes is to discover the smallest fraction of components, which when encrypted produce sufficient distortion in the decoded video/audio. For example, speech encryption of the spectral envelope and the gain codebooks decreases the intelligibility of semantic content, diminishes any possibility of gender identification or its role in discriminating between speech and silence[14]. We first present an overview of some of the design principles in PE schemes.

2.1. Design Principles in Partial Encryption Schemes

The compressed image or video, prior to partial encryption is broken into two main components, the perceptually significant set or the sensitive set X_{PS} and other components X_{OC}. It is very important that there be a strong connection between the two components and yet one should not be able to derive one from the other. Figure 1, shows a stream X broken down into a number of components that are connected via this dependence graph. The nature of the partitioning and selection of components to be encrypted is often the challenge of the partial encryption process for the following reasons enlisted below.

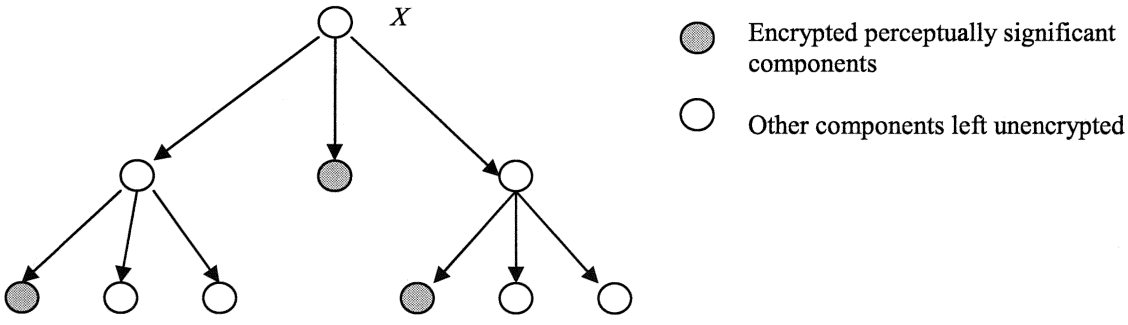

Figure 1. Partial encryption of a compressed stream

1. **Spatio-temporal correlation between objects:** Digital images and video are highly correlated processes. Most video compression schemes such as MPEG-2, 4, exploit the spatio-temporal redundancy in the video scene, to

reduce the bit rate. Video and most of the recent image compression schemes have a layered (or hierarchical structure), in which certain objects (or frames) are derived from a base object. Hence, by encrypting the base objects, one can obscure a large portion of the multimedia. This layered structure also facilitates transcodability without complete stream decoding. If we represent the amount of visual information in a image or video clip X using the function $V(.)$, we can write,

$$V(X_{PS} \,||\, X_{OC}) = V(X) \tag{1}$$

$$V(E_K(X_{PS}) \,||\, X_{OC}) \neq V(X) \tag{2}$$

For example, in Ref. 15, encrypting I-frames of a MPEG-2 video stream renders the other derived frames P and B relatively useless. But there are still sufficient numbers of Intra-coded-blocks in the P frames, which can be used to derive information regarding the parent frame. On the other hand if I frames, intra-coded blocks in the P frames and the motion vectors were encrypted, this would increase the difficulty for the attacker to reconstruct the I-frame from the unencrypted frames.

2. **Statistical independence:** X_{PS} and X_{OC} must be ideally statistically independent It is well known that E[sign(DCT-AC-coefficients)] = 0 and are independent of the amplitudes. This means, knowledge of the amplitudes will not reveal any information regarding the sign bits. So the sign bits alone can be encrypted using a pseudo random sequence to produce some distortion without compromising on secrecy and compression efficiency. The VEA scheme based on this principle[16], though computationally very efficient, does not produce sufficient obscurity of the image.

3. **Impact on compression:** In some cases, encryption of X_{PS} may affect the overall compression ratio, based on the domain in which the encryption is performed. Even if a fraction of the input to the entropy coder is encrypted, then the bit rate increases due to a compromise on statistical redundancy for secrecy. For example in Ref. 16, coefficients within the block are shuffled after transform coding and quantization. This affects the runlength of zeros and the overall compression ratio.

Many schemes have been proposed to address this problem. The VEA algorithm[16], as mentioned earlier encrypts only the sign bits that has virtually no impact on compression since the sign bits form an IID sequence. Cheng et al[18] proposed a technique based on embedded coding in which the significance map connecting the perceptual significant coefficients in the wavelet domain were encrypted. It was shown here that although the significance map forms a substantially small component of the transmitted image, encrypting these maps produces sufficient distortion and it is virtually impossible to reconstruct the map from the unencrypted components. Since the size of these significance maps are very small, these schemes are very efficient.

2.2. Merging decryption and fingerprinting

Our objective is to incorporate fingerprinting in this partial encryption framework by designing different decryption keys $\{K_{D1}, K_{D2}...\}$ for restricting access and tracking users $\{U_1, U_2...\}$ such that when the secure multicast stream is decrypted an imperceptible fingerprint is embedded in the content. The relative entropy between the encryption and decryption keys, $H(K_E/K_{Di})$ bits/channel usage gives an approximate measure of the fingerprint payload which is a function of the correlation between the encryption and decryption keys.

As shown in Figure 2, a compressed stream C is partitioned into different components based on some of the conditions discussed in Section 2.1. The perceptually significant components, denoted as set S are encrypted. At the receiver based on the decryption key, all but a segment of the perceptually significant set is decrypted. The segment left encrypted is denoted as F_i and has a unique signature. The number of uniquely distinguishable fingerprints that can be embedded depends on factors such as the size and nature of the significance set (which depends on the transform domain, thresholds used for the extraction of coefficients etc).

It may seem practical to select the smallest possible set for encryption in a wireless application, but then this does not provide sufficient scope for embedding an imperceptible and robust fingerprint at the receiver. If the sensitivity constraints are relaxed while creating the significance set, the diversity of the fingerprint can be increased without compromising on imperceptibility. Thus, one main caveat with this architecture is that watermark robustness must be

traded for computational complexity of the encryption process. Another problem in this framework is that since, $D_{K1}(Y) = X_1 \cong D_{K2}(Y) = X_2$ (for imperceptibility), the ciphertext cannot be a complex and non-linear function of the encryption key set as in DES. Consequently fingerprinting softens the encryption process.

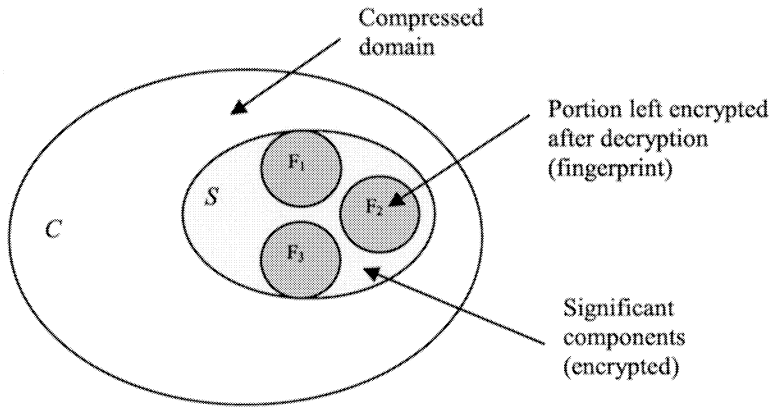

Figure 2. Illustration of the joint fingerprinting process

In source based fingerprinting schemes[3-6], the process of fingerprinting is completely decoupled from the encryption/decryption process. The consequence is that separate keys can be associated with watermarking and encryption. The key for watermark embedding could be used to identify the location of the objects (or group of coefficients) where the watermark bits are concealed and no part of this information is revealed to the receiver.

However in the joint fingerprinting and decryption framework (JFD) framework, the decryption key(s) in conjunction with the encryption key is used not only to embed the fingerprint but also to descramble the components. Hence, from the point of view of the attacker, the search space for the fingerprint is restricted to the fraction of components that are encrypted (which is small in most partial encryption schemes). In order to embed a robust fingerprint the decryption keys have to be designed along with the encryption key. The consequence of using a single key is that if the designer decides to spread the fingerprint over a large number of components to increase robustness, then the computational complexity associated with this soft encryption increases and so does the key size.

2.3. Potential Applications for JFD

Asymmetric embedding: Asymmetricity in watermark embedding (function of two different keys) coupled with encryption adds another dimension to content protection since the trust model with watermark embedding is now distributed across two or more individuals. The asymmetric nature associated with embedding makes this framework a good choice for creating audit tails without having to trust each one of the individual nodes with the embedding process. If F is the original watermarked document and if each user in the chain modifies this fingerprint in controlled and unique fashion through unique encryption/decryption key pairs, we would get a trail $F \rightarrow F1 \rightarrow F2 \rightarrow F3 \ldots$ The manner in which the manipulation can be done depends on the nature of the algorithm. For example, each user could be forced to erase a different portion of F. Hence, if the set difference *F(source) – F(recovered)* is unique for each subset of nodes, then the trail can be detected reliably. A fingerprinting scheme based on those lines was Watercasting[11], in which the intermediate nodes (or routers) drop unique segments of a collection of redundant media clips en-route to the receiver (or leaf node). The end-to-end distortion was kept minimal by introducing sufficient redundancy to minimize errors due to packet loss. An application is the embedding of a fingerprint within a sensitive e-record (such as a health record), each time the record is decrypted and viewed. If the record is forwarded, it becomes a problem of progressive embedding of watermarks within a document, which raises issues related to robustness and imperceptibility as the trail grows in length. In such a framework,

- Fingerprints must be robust to re-encryption and re-compression.
- The reliability associated with successful fingerprint detection will depend on the number of mutually orthogonal trail-markers that can be reliably embedded within the document. In the JFD framework, this will

be a function of the encryption and decryption key-pairs that are distributed to each pair of nodes. Applicability of JFD to such a framework is still under investigation.

Facilitates secure multicasting: This makes it very useful (i.e. bandwidth and computationally efficient) for secure audio and video distribution[13,19]. A single stream can be encrypted and sent to various receivers, who are given different decryption keys designed based on a single encryption key. The fingerprint is embedded during the process of decryption. The challenges here are to control the correlation between these key sets so that the decrypted/fingerprinted copy is collusion resistant and also robust to incidental distortions.

In this paper we focus on the simpler multicast-fingerprinting problem in which an encrypted document broadcast by a node is fingerprinted when n receiving nodes decrypt it.

3. SIMPLE ANALYTICAL MODEL FOR JFD

We present an analytical model to illustrate some of the tradeoff issues in the JFD architecture. The model is based on the encryption of the sign-bit plane of perceptually significant coefficients that comprises of a sequence of binary IID random variables, $X \in \{-1,1\}$. The perceptually significant coefficients are first extracted and their sign bits are encrypted by multiplying a pseudo random key sequence $K_E \in \{-1,1\}^n$. The encryption process, decryption/watermarking and incidental distortions are modeled as a cascade of three Binary Symmetric Channels (BSCs) *C1, C2* and *C3* with cross over probabilities p_1, p_2 and ε respectively, as shown in Figure 3. The cross over probabilities of the 1st and 2nd channel is a function of the correlation between the encryption and decryption keys. The decryption keys distributed to the receivers are $K_{Di} \in \{-1,1\}^n$, $i = 1 \ldots N_f$, where, N_f is the number of users (or fingerprints). The decryption key K_{Di} is designed such that $\langle K_E, K_{Di} \rangle \geq r \in [0,1]$, so that the fingerprint is imperceptible.

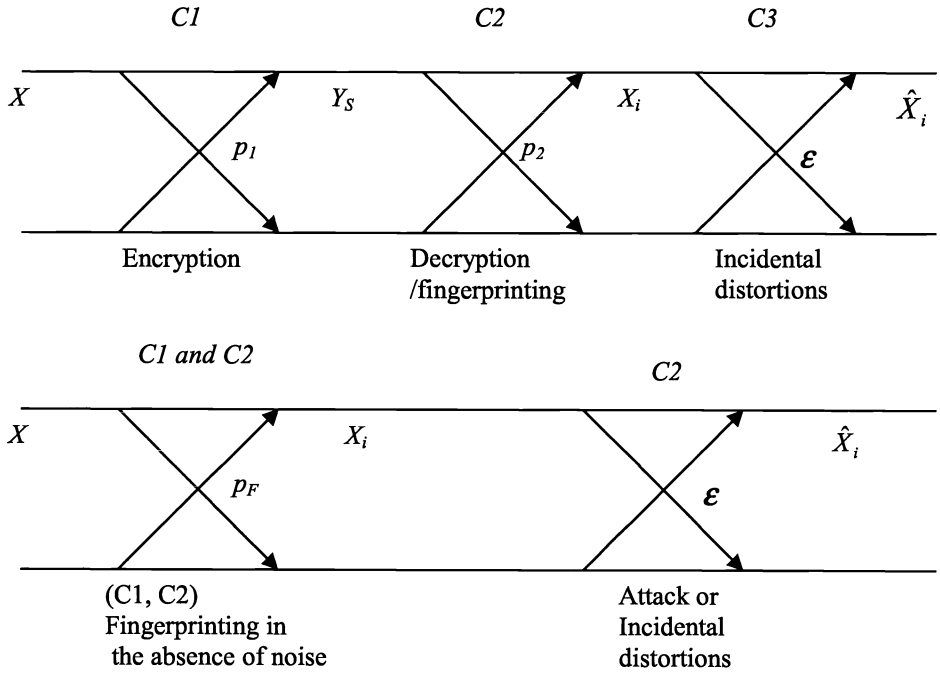

Figure 3. BSC model for JFD

In the absence of C3 or any distortions, the fingerprinted sequence is given by,

$$X_i = (K_{Di} \bullet K_E) \bullet X = F_i \bullet X \tag{3}$$

and with C3,

$$\hat{X}_i = K_C \bullet X_i = K_C \bullet K_E \bullet K_{Di} \bullet X = \hat{F}_i \bullet X \tag{4}$$

where, '•' represents the overloaded multiplication operator, [-1, 1] • [1, -1] =[-1.1, 1.-1] = [-1,-1]. The sequence K_C is a function of the crossover probability of the channel *C3*. For perfect secrecy, K_E must be a one-time pad, which implies $p_1 = 0.5$. It can also be shown that, $p_F = (1-r)/2$. As per Equation (4), it is important for the sign bit plane X to be a sequence of IID random variables. Otherwise this increases the chances of fingerprint detection as well as erasure. This is one of the reasons why the fingerprint was not embedded in the most significant bit planes in the algorithm discussed in Section 4 below. The sign bits of DCT coefficients form an IID sequence and hence a perfect cover for the fingerprint. Also, the sign bits are robust to recompression if only the highly perceptually significant coefficients are extracted. The LSB could also be a good cover for the fingerprint due to the randomness introduced by the quantization process in the lower bit planes, but the embedded fingerprint would be susceptible to lossy recompression (large ε). The fingerprint capacity (without *C3*) is given by,

$$C_{Fi} = H[F_i/(X_i, K_{Di})] = H[K_E / K_{Di}] = H[(1-r)/2] \tag{5}$$

Any intentional or unintentional manipulation of the image reflects as an error in the fingerprint code, which has been modeled as the channel *C3*. The error probabilities can be derived empirically, by subjecting the image to a variety of attacks. The corresponding fingerprint capacity is,

$$C_{Fi} = \min[H[(1-r)/2],\ H(\varepsilon)] \tag{6}$$

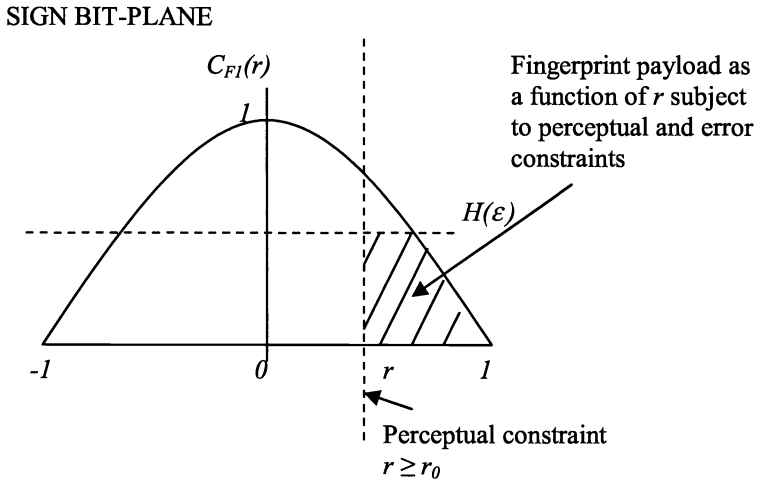

Figure 4. Covert channel capacity

Thus the maximum fingerprint payload in an n bit encryption stream will be approximately nC_{Fi} bits. Figure 4 shows the maximum payload (Equation. (6)) which can be embedded in the bit plane owing to perceptual constraints and also channel errors introduced because of recompression, intentional manipulation etc. The perceptual constraint imposed by r_0 can be by changing the thresholds used for creating the significance set. But this increases the size of the decryption key stream and also the computational complexity of the encryption process. The process of generating decryption keys is shown in Figure 5. To meet the imperceptibility requirement, the decryption key streams must be generated which meet the correlation constraint,

$$\langle K_E, K_{Di} \rangle \geq r_0, \quad i = 1,2,...N_f \tag{7}$$

The fingerprint is channel coded and modulated with a pseudorandom sequence S_i. The length of the sequence S_i is a function of the correlation coefficient r. The code and modulated sequence F_i is then merged with the encryption key K_E as a function of the secret key K_{fi}. Intuitively one can see that the correlation r affects the payload of the fingerprint (k bits) and the sequence length m. The decryption keys are then unicast to the receiving nodes.

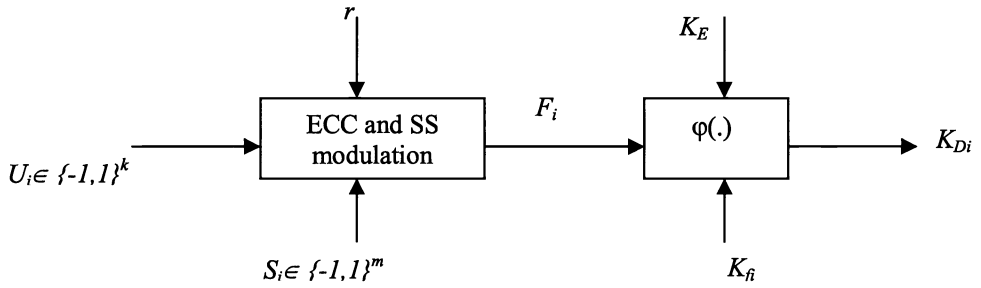

Figure 5. Decryption key generation process

4. PROPOSED ALGORITHM

The proposed scheme based on the JFD architecture discussed in Sects. 2 and 3 is used for encrypting and fingerprinting JPEG compressed images. Two levels of encryption are performed: sign bit-plane encryption (L1) and AC coefficient permutation (L2). L1, which is a weak encryption process serves as a cradle for fingerprinting. A set of unique decryption keys are derived from the encryption key which are used to encrypt the sign bit plane. L2 encryption is essential to produce sufficient obscurity in the image to eavesdroppers. The steps in the encryption process have been summarized below.

Step 1: Extraction of perceptually significant coefficients for L1 encryption

$AC\_SIG = \phi$

For all the quantized 8X8 blocks, B_j, $j = 1...N\_blocks$,

$T_L(j) = \frac{|AC|_{\max}}{2^{k_1}}, T_U(j) = \frac{|AC|_{\max}}{2^{k_2}}$, where $k_1, k_2 \geq 0$

If $|x_{Bj}(i)| \in [T_L(j), T_U(j)]$, where $i = \{2...63\}$, then $AC\_SIG = AC\_SIG \cup \{x_{Bj}(i)\}$.

Step 2 (a): Sign bit encryption (L1)

The extracted sign-bit plane, $X_{SIGN} = sign(AC\_SIG\_values)$

The encrypted bit plane, $Y_S = K_E \bullet X_{SIGN}$

where, $K_E \in \{-1,1\}^{length(AC\_SIG)}$

Step 2 (b): Decryption key generation and fingerprint embedding

$\langle K_E, K_{Di} \rangle \geq r_0$, where the threshold r_0 is determined based on imperceptibility constraints from empirical results. The number of positions that differ in the encryption and decryption sequences is calculated as $n_d = \lfloor (n-n.r)/2 \rfloor$, where $\lfloor \rfloor$ represents the floor function. The encryption key is modified with a pseudorandom sub-sequence F_i of length $n_S = 2n_d$ to generate the decryption key K_{Di}. The presence of the sequence F_i in the decryption key generates a unique signature for each receiver during the decryption process.

Step 3: Permutation (L2)

Only the largest AC coefficients from each block are permuted. This is a symmetric encryption process and hence no distortion is incurred after decryption at the receiver. The L2 permutation key and the decryption key stream K_{Di} are unicast to the receivers. At the receiver, the processes are reversed and L2 decryption is followed by L1 decryption/fingerprinting with key K_{Di}.

4.1. Simulation Results and Discussions

Table 1: Parameters for the encryption and fingerprinting of a 256 X 256 LENA image

Thresholds	$k_1 = 0$, $k_2 = 2$ (for L1 encryption) Highest AC coefficients selected from each block for L2 encryption
Correlation between keys (r)	0.95
Number of coefficients in the significance set of L1	10327
Number of coefficients in the significance set of L2	12450
Total coefficients extracted	12450
N_bits	9
$\lvert AC_{max} \rvert$ (maximum value of ac coefficient)	278
Number of unique AC magnitude levels	169

4.1.1. L1 encryption and fingerprinting

Based on the parameters in Table.1, the PSNR values of the L1 encrypted and fingerprinted/decrypted images shown in Figure 6. were (23.38dB and 38.37dB respectively. It can be seen in Figure 6(b), that distortion in the L1-encrypted image is not significant, because of which another level of encryption was introduced.

(a) (b) (c)

Figure 6. Original, L1-encrypted and decrypted/fingerprinted images

The effect of correlation on the perceptibility of the fingerprint is shown in Figure 7. For large correlations the fingerprint is almost imperceptible as can be seen from the plot. However, this comes at a price of a reduction in fingerprint payload and robustness. The effect of lossy recompression on fingerprint robustness is shown in the Figure 8 (a,b,c). A peak in the cross correlation between the sign-bit plane retrieved from the significant set and the fingerprint sequence F_i, indicates the presence of a fingerprint sequence concealed within the sign bit plane. As shown in Figure 8 (a,b,c), the fingerprint is detectable for $Q = 100$ (no attack) and $Q = 70$, but becomes virtually undetectable for $Q \geq 50$, since a large fraction of the AC coefficients are zeroes. By decreasing the correlation the spread of the fingerprint can be increased and consequently the robustness, but this introduces some perceptual distortion in the image. This can be seen in Figure 9 wherein although the fingerprint is successfully detected, it leaves some perceptible block artifacts. The PSNR value in this case has dropped to 35.81dB.

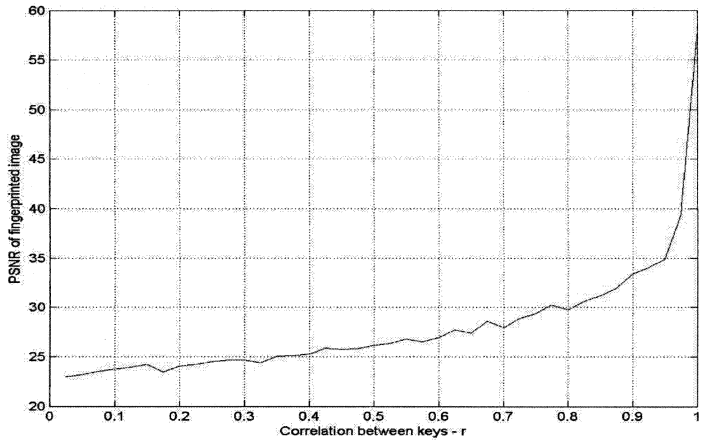

Figure 7. PSNR as a function of correlation r between the encryption and decryption keys.

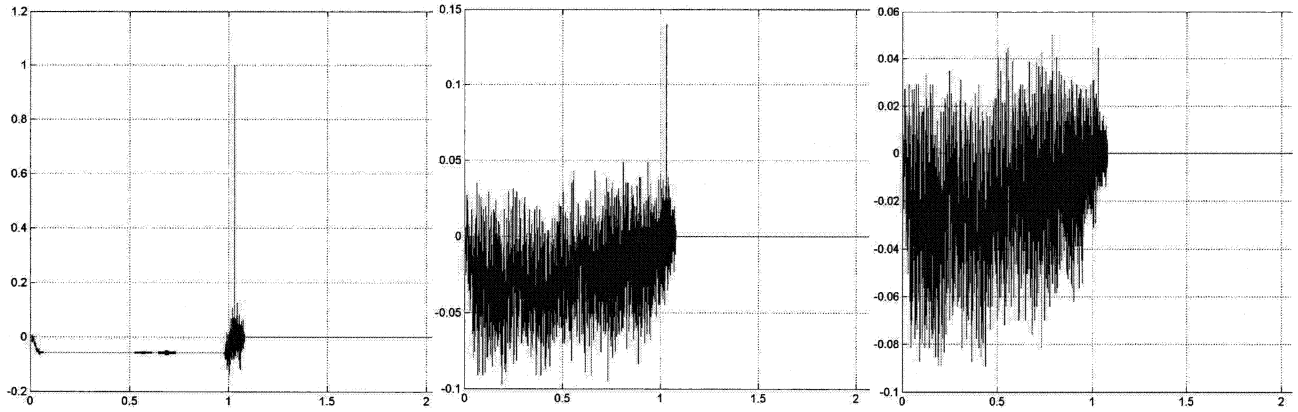

Figure 8. Fingerprint detection results, (a) $r = 0.95$, $Q = 100$ (no attack), (b) $r = 0.95$, $Q = 70$, (c) $r = 0.95$, $Q = 50$.

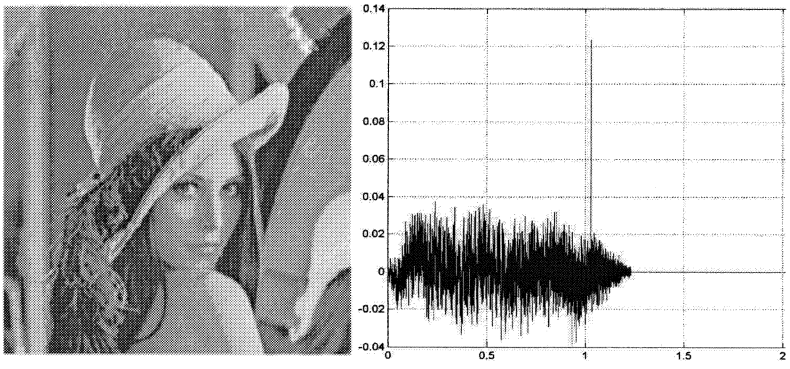

Figure 9. Fingerprinted image with $r = 0.8$ for $Q = 50$ and the corresponding fingerprint detection results.

4.1.2. L1 + L2 encryption

One can observe a significant distortion when both substitution (L1) and permutation (L2) ciphers are executed. The PSNR of the encrypted image shown in Figure 10 was 18.23dB which is much lower than the PSNR of the L1 encrypted image (23.38dB) in Figure 6(b).

Figure 10. L1 + L2 encrypted image

The permutation operation in L2 has very little impact on the bit rate, since only highest AC coefficients from each block are extracted for permutation. In the JPEG format, originally proposed by Wallace et al.[20], every non zero AC coefficient is represented along with the run length of zero valued AC coefficients which precede it in a zig-zag sequence. This is represented by two symbols SYM-1: (RUN LENGTH, SIZE) and SYM-2: (AMPLITUDE). A permutation of coefficients is equivalent to shuffling the indices in Table-2[20], which has minimal impact on compression ratio, provided the shuffling is done in a controlled fashion. It is highly probable that larger run-lengths are usually followed by coefficients with small amplitudes. Because of this, random shuffling may result in suboptimal Huffman code assignment and a consequential increase in bit rate. In this algorithm we mitigate this problem by controlling the threshold values used for selecting the significant coefficients.

5. CONCLUSIONS

In summary, through a simple analytical model and an algorithm, we have illustrated some of interesting design challenges in the JFD framework. Selection of highly perceptible components reduces the computational burden associated with encryption, since the size of the significance set is small, but this comes at a price of compromising either fingerprint robustness or payload. To meet the imperceptibility constraint, the ciphertext (i.e. the encrypted image components) cannot be a complex and non-linear function of the encryption key. Thus merging fingerprinting with decryption softens the encryption process because of which we have implemented a two level encryption scheme to produce sufficient distortion in JPEG images: L1, based on an asymmetric key design in which the sign bit plane of the perceptually significant AC coefficients is encrypted, while, L2 is a symmetric key permutation cipher. Fingerprinting is performed at the L1 level. The encryption process is format compliant and can be integrated with the JPEG compression scheme after the quantization stage without compromising significantly on the compression efficiency. Our next step would be to investigate other ways of improving the fingerprint capacity within the partial encryption framework. Designing a fingerprint robust to linear and non-linear collusion attacks within the JFD framework would be another challenge. We are currently in the process of developing more sophisticated analytical models for JFD with the objective of developing some general design guidelines for embedding fingerprints across a broader class of encryption schemes.

REFERENCES

1. C.-Y. Lin, M. Wu, J. A. Bloom, M. L. Miller, I. Cox, and Y. M. Lui, "Rotation, scale, and translation resilient public watermarking for images," *SPIE Security and Watermarking of Multimedia Contents II, EI'00*, San Jose, CA, 2000.
2. Ingemar J. Cox, Matthew L. Miller and Jeffrey A. Bloom, *Digital Watermarking*, published by Morgan Kaufmann, 2001.
3. Cox, I.J.; Kilian, J.; Leighton, F.T.; Shamoon, T, "Secure spread spectrum watermarking for multimedia", *IEEE Transactions on Image Processing*, vol. **6**, no. 12, p. 1673- 87, 1997.

4. A. Piva, M. Barni, F. Bartolini, and V. Cappellini, "DCT-based watermark recovering without resorting to the uncorrupted original image", *Proceedings of 4th IEEE International Conference on Image Processing ICIP'97*, Santa Barbara, CA, USA, October 26-29, 1997, Vol **I**, pp. 520-523.
5. P Fernando Pérez-González, Juan R. Hernández, and Félix Balado, "Approaching the capacity limit in image watermarking: A perspective on coding techniques for data hiding applications", *Signal Processing, Elsevier*, vol. **81**, no. 6, pp.1215-1238, June 2001.
6. S. Baudry, J.F. Delaigle, B. Sankur, B. Macq, H. Maitre, "Analyses of Error Correction Stratgies for Typical Communication Channels in Watermarking, Signal Processing", *Signal Processing, Elsevier*, vol. **81**, no. 6, pp. 1239-1250, June 2001.
7. W. Trappe, M. Wu, Zhen Wang, K.J.R. Liu, "Anti-collusion Fingerprinting for Multimedia", *submitted to IEEE Trans. on Signal Processing*, Dec. 2001, revised July 2002.
8. D. Boneh and J. Shaw, "Collusion secure fingerprinting for digital data", *IEEE Trans. on Information Theory*, vol. **44**, pp.1897—1905, Sept 1998.
9. Z.J. Wang, M. Wu, W. Trappe, and K.J.R. Liu, "Anti-Collusion of Group-Oriented Fingerprinting", *IEEE International Conference on Multimedia & Expo (ICME'03)*, Baltimore, MD, July 2003.
10. K. Su, D. Kundur and D. Hatzinakos, "A Novel Approach to Collusion-Resistant Video Watermarking", Security and Watermarking of Multimedia Contents IV, E. J. Delp and P. W. Wong, eds., *Proc. SPIE* (vol. 4675), 12 pages, San Jose, California, January 2002.
11. Ian Brown, Colin Perkins, and Jon Crowcroft.,"Watercasting: Distributed watermarking of multicast multimedia", *First International Workshop on Networked Group Communication (NGC99)*, 1999.
12. Paul Judge, Mostafa Ammar, "WHIM: Watermarking Multicast Video with a Hierarchy of Intermediaries", *Proceedings of NOSSDAV 2000*, Chapel Hill, NC, June 2000.
13. R.J. Anderson and C. Manifavas, "Chameleon -- A New Kind of Stream Cipher", *Proc. Fourth Workshop on Fast Software Encryption*, pp. 107-113, January 1997.
14. A. Servetti, J.C. De Martin, "Perception-Based Partial Encryption of Compressed Speech", *IEEE Transactions on Speech and Audio Processing*, vol. **10**, no. 8, pp. 637-643, Nov. 2002.
15. I. Agi and L. Gong, "An empirical study of secure MPEG video transmissions", *Internet Society Symposium on Network and Distributed System Security*, Feb. 1996.
16. Lei Tang, "Methods for Encrypting and Decrypting MPEG Video Data Efficiently", *Proceedings of the ACM Multimedia 96*, pp. 219-229, Nov, 1996.
17. Changgui Shi and Bharat Bhargava, "A Fast MPEG Video Encryption Algorithm", *Proceedings of the 6th ACM International Multimedia Conference*, Sep, 1998.
18. Cheng, H. and Li, X.,"Partial Encryption of Compressed Images and Videos", *IEEE Transactions on Signal Processing*, vol. **48,** no. 8, pp. 2439-2451, 2000.
19. D. Kundur and K. Karthik, "Digital Fingerprinting and Encryption Principles for Digital Rights Management", . *Proceedings of the IEEE Special Issue on Enabling Security Technologies for Digital Rights Management*, (full paper) to appear mid 2004.
20. G. K. Wallace, "The JPEG still picture compression standard", *IEEE Transactions on Consumer Electronics*, vol. **38**, no. 1, pp. 18-34, Feb. 1992.