The 9th International Symposium on Virtual Reality, Archaeology and Cultural Heritage
VAST (2008)
M. Ashley, S. Hermon, A. Proenca, and K. Rodriguez-Echavarria (Editors)

A Peer to Peer Network Environment for Optimized Digital Rights Management of Digital Cultural Heritage

D. Tsolis [1]

[1]Department of Computer Engineering and Informatics, University of Patras

Abstract

As a general protection measure for copyright violations through digital technologies including peer to peer (P2P), copyright owners often uses Digital Rights Management (DRM) techniques to encrypt and watermark content or otherwise restrict access, totally blocking digital content to be accessed through the Internet and the P2P infrastructure. This paper claims that DRM and P2P can be quite complementary. Specifically, a P2P infrastructure is presented which allows broad digital content exchange while on the same time supports copyright protection and management through DRM technologies.

Categories and Subject Descriptors (according to ACM CCS): H.5.1 [Multimedia Information Systems]: evaluation methodology

1. Introduction

Peer to Peer networking is supported by suitable software which enables a computer to locate a content file (text, image, video, sound, software etc.) on another networked device and copy the encoded data to its own hard drive. P2P technology often is used to reproduce and distribute copyrighted content without authorization of rights owners. Except for digital music and video the P2P infrastructure is also used to make and distribute illegal copies of digital cultural content. For this reason the short history of P2P technology and software has been one of constant controversy by many in the content industry. In the Cultural Heritage area the content owners are feeling even more threatened by the broad and unregulated exchange of digital content in P2P environments [cstb].

As a general protection measure for copyright violations through digital technologies including P2P, copyright owners often uses Digital Rights Management techniques to encrypt and watermark content or otherwise restrict access, totally blocking digital content to be accessed through the Internet and the P2P software infrastructure.

This paper claims that Digital Rights Management and P2P can be quite complementary. Specifically, a P2P network infrastructure is presented which allows broad digital content exchange while on the same time supports copyright protection and management through DRM technologies. In brief, the platform is functioning mainly for digital images of cultural heritage and is tracking all the watermarked image files which are distributed and copied through the P2P network. The challenge is the algorithmic complexity of detecting multiple watermarking keys in the P2P network effectively and quickly, especially when thousand of image files are concerned. This is managed by an optimization detection algorithm which allows effective watermarking key detection in optimal P2P hops.

Equivalent systems, which combine DRM and P2P technologies do not yet exist in practice but only in theory. Certain methodologies and strategies have been proposed for exploiting P2P technologies in DRM and vice versa [einhorn]. The proposed system is setting a new basis for the close cooperation of the the two different scientific areas of DRM and P2P aiming at exploiting the distributed computing nature of P2P networks for efficient digital rights protection and management.

2. DRM Protection - The Watermarking Algorithm Keys

In this section the DRM protection part of the P2P infrastructure is presented which is mainly based on a watermarking

algorithm for digital images which produces the correspodent watermarking keys distributed within the P2P environment.

2.1. DRM and Watermarking

The DRM system's main objectives are to provide an appropriate information infrastructure which supports rights management for the digital content and for the transactions taking place and on the same time protects the copyright of the digital images though robust watermarking techniques.

The watermarking techniques are playing a very important role in such systems mainly because they provide the protection means for proving the identification of the copyright owner and detecting unauthorized use of digital content [wayner]. Towards this functionality, watermarking algorithms are casting keys to the digital content (in most of cases invisible keys) which when detected prove the copyright ownership of the digital content [cappellini].

In case of digital content transactions a very large number of digital images are being exchanged through networks and the Internet for which the legality of their future use is highly unprobable. The situation is even more difficult in P2P network infrastructures through which digital content is being exchanged based on specialized stand alone applications which exchange digital files of all kinds (and not only images).

A proposed solution is to apply a watermarking algorithm which produces sufficient information which is distributed to the P2P nodes. This information consists mainly of the watermkarkring key and other data relating to the digital image itself.

2.2. The Watermarking Algorithm - Generating Keys

Generally, a watermark is a narrow band signal, which is embedded to the wide band signal of a digital image [randall]. In our case spread Spectrum techniques are being used and are methods by which energy generated at one or more discrete frequencies is deliberately spread or distributed in time or frequency domains.

In particular, this technique employ pseudorandom number sequences (noise signals) to determine and control the spreading pattern of the signal across the allotted bandwidth. The noise signal can be used to exactly reconstruct the original data at the receiving end, by multiplying it by the same pseudorandom sequence: this process, known as "de-spreading", mathematically constitutes a correlation of the transmitted pseudorandom number sequence with the receiverŠs assumed sequence. Thus, if the signal is distorted by some process that damages only a fraction of the frequencies, such as a band-pass filter or addition of band limited noise, the encrypted information will still be identifiable. Furthermore, high frequencies are appropriate for rendering the watermarked message invisible but are inefficient in terms of robustness, whereas low frequencies are appropriate with regards to robustness but are useless because of the unacceptable visual impact [cox].

In our case, the embedding of a robust multibit watermark is accomplished through casting several zero-bit watermarks onto specified coefficients. The image watermark, a random sequence of Gaussian distribution in our case, is casted multiple times onto the selected coefficients preserving the same sequence length but shifting the start point of casting by one place.

Actually the final watermark that will be embedded into the image is not a single sequence but many different sequences generated with different seeds. These sequences will be casted, one after the other, on the mid coefficients of the image, using the additive rule mentioned above and begging from successive starting points. If all sequences where to be casted, beginning from the same starting point, then, besides the severe robustness reduction resulting from the weak correlation, the possibility of false positive detector response would dramatically increase, since every number that has participated as a seed during the sequence generation procedure, will be estimated by the detector as a valid watermark key. Shifting the starting point by one degree for every sequence casting ensures that the false positive rate will remain in very small level due to the artificial desynchronisation introduced. Every single random sequence of Gaussian distribution is generated using a different number as the seed for the Gaussian sequence generator. It is important to differentiate the sequences in order not to mislead the detection mechanism, since it is based on the correlation between the extracted sequence and the sequence produced with the watermark key.

The watermark key is responsible both for the generation of the first sequence and the construction of a vector, containing the rest of the numbers that will serve as the corresponding seeds. The placement of several Gaussian sequences into the image content can model, under specific conventions, a multibit watermark. The detection of a zero-bit watermark is interpreted as if the bit value of the specified bit is set to one. On the contrary, failure of the detector to detect the zero-bit watermark leads to the conclusion of a zero bit value. Thus, in order for a message to be casted into the image content, it is initially encoded using the binary system and applied afterwards in the sense of zero-bit watermarks using the embedding mechanism and according to the derived bit sequence.

Some important remarks regarding the novelty of the proposed schema are addressed below.

Data payload: The reason that most of the proposed robust watermarking systems are zero-bit, is highly related to the data payload. Data payload is the amount of information encoded into the image during the watermark procedure. In other words, it is the number of coefficients modified according to the additive rule. The performance of the correlation

function adopted by the detector is increased when a strong statistical dependency is present. On the other hand, the statistical dependency requires a significant sequence length in order to fulfill the requirements of the correlation function. In addition, the position and the amount of coefficients modified, affects directly the resulting image quality. This is one of the most important tradeoffs that the designer of a watermarking system has to balance.

Casting multiple sequences will maximize the problem of image distortion. In that sense, the maximum number of bits allowed for encoding the watermark message is crucial. In the proposed scheme a total number of 16 bits were selected. The first bit indicates the existence of a watermark. If the response is positive the detector continues with the following zero-bit watermarks, otherwise the mechanism outputs a negative response. This is a useful shortcut saving the detector of valuable time and processing power. The second bit serves as a flag important for the decoding operation. The role of this bit flag is described in detail in the following paragraph. The next 14 bits are dedicated to the encoding of the watermark message. Under the aforementioned conventions the system is capable of embedding 2^{14} different messages.

Seed Vector Generation: The watermark key is a positive integer value playing a vital role in the overall watermarking procedure. It corresponds to the private information that must be shared between the embedder and the detector of the watermark. One of the basic principles of private watermarking is that the encryption of the information to be embedded is performed according to a private key. Thus, if an image is watermarked using a specified key, it is impossible for the detector to detect the watermark unless provided with the same key. The encryption is accomplished by using the private key as the seed for the pseudorandom sequence of Gaussian distribution generator. In our case, there is the necessity of 15 extra numbers, one for each sequence. Thus, the private key except from its basic operation as a pseudorandom generator seed is also used as the seed for producing a vector containing 15 numbers. It is important for every private key to produce a different vector of numbers, in order to avoid undesirable statistical dependencies between different watermarks. A pseudorandom generator provided by any compiler is capable of applying this one-way relationship between the private key and the produced vector of numbers.

Flag bit operation: Under the convention, that for every one-bit-value we cast a zero-bit watermark and for every zero-bit-value we don't do anything except moving to the next starting point, the number of zero-bit watermarks to be casted is dictated by the bit sequence. It is obvious that a bit sequence containing only a single one-bit-value is preferable from a sequence consisted of 14 aces. Both for, processing power and watermarkŠs imperceptibility purposes, a bit reversal trick is required for optimizing the embedder's performance.

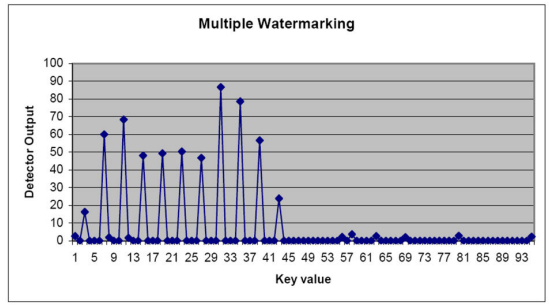

Figure 1: *Multiple Watermarking Keys per Image*

Thus, after acquiring the binary representation of the message, a counter scans the bit sequence counting the zeros and the aces. If the number of aces is grater than the number of zeros a bit reversed sequence is generated. The zero-bit watermarks casting is now performed according to the newly generated sequence. In that case, the flag bit is set to one serving as an indicator to the detector that the extracted sequence is bit-reversed. As a consequence, the decoder, equipped with the appropriate information, can easily decode a message represented by 14 aces binary sequence, even though the embedder had casted only two zero-bit watermarks. The benefit of using the specified trick is that even though a 16-bit watermark is supported, we only need to cast 8 zero-bits watermarks in the worst case.

The detector used in the proposed information system reveals the existence of 11 watermarks. Three of them correspond to the three zero-bit schemes while the rest 8 positive responses are used for the encoding of the fingerprint. The detector has succeeded in detecting all eleven watermarks without any confusion or misleading, resulting in a capability of facilitating proof of ownership, copy control, digital signature and transaction tracking at the same time [house].

2.3. Intermediate Conclusions

In this section a watermarking algorithm has been presented which is robust enough to facilitate copyright protection and management for the digital image of cultural heritage while at the same time produces sufficient information which is distributed and sotred to the P2P nodes. This information consists mainly of the watermkarkring key.

Taking into consideration that for each digital image a set of watermarking keys are being used for copyright protection, the next step towards an efficient P2P environment which supports digital rights management is to use these keys as an information for retrieving the copyright status of each image transacted through the P2P network. For this

reason, the watermarking keys are being stored in the independed network Peers. The copyright owner can use the watermarking key as a query information to track down its digital images and their use. The issue is how quickly and efficiently the Peer that contains the under inspection key is being located taking into account that thousands of digital images could exist in the P2P network and multiple watermarking keys could exist in a digital image. The solution proposed is a scalable and robust data indexing structure based on a Nested Balanced Distributed Tree (NBDT).

3. Overview of the NBDT network

NBDT provides a tree-like structure for the P2P network upon which watermarking key-based searching can be performed. In terms of bandwidth usage, searching scales very well since no broadcasting or other bandwidth consuming activities take place during searches. Since all searches are key based there are two possibilities: either (a) each host implements the same algorithm, that translates a keyword to a binary key or (b) another service provides the binary key. This service accepts keyword based queries and can respond with the corresponding key. The second approach is more precise. It is also possible to use a more centralized implementation for such a service. From now on we assume that the key is available. This section describes an algorithm for the first case.

The structure was built by repeating the same kind of BDT tree-structure in each group of nodes having the same ancestor, and doing this recursively. This structure may be imposed through another set of pointers. The innermost level of nesting will be characterized by having a tree-structure, in which no more than two nodes share the same direct ancestor. The figure 2 illustrates a simple example (for the sake of clarity we have omitted from the picture the links between nodes with the same ancestor). Thus, multiple independent tree structures are imposed on the collection of nodes inserted. Each element inserted contains pointers to its representatives in each of the trees it belongs to.

Let σ an initial given μ sequence of w-bit keys belonging in universe K=[$0, 2^w$-1], where μ an unknown density. At initialization step we choose as peer representatives the 1st key, the $lnK^s t$ key, the $2lnK^s t$ key and so on, meaning that each node with label i (1<i<N) stores ordered keys that belong in range [(i-1)lnK,..ilnK-1], where N=K/lnK the number of peers. Note that during update operations; it is not at all obvious how to bound the load of the N peers, since new w'-bit keys with w'>w may be appeared in the system and K must exceed. For this purpose we will model the insertions/deletions as the combinatorial game of bins and balls presented in [kaporis 2003]: Modelling the insertions/deletions of keys in this way, the load of each peer becomes Θ(polygonN) in expected case with high probability. Obviously, peers' representatives early described have also been chosen according to this game. We also assume that

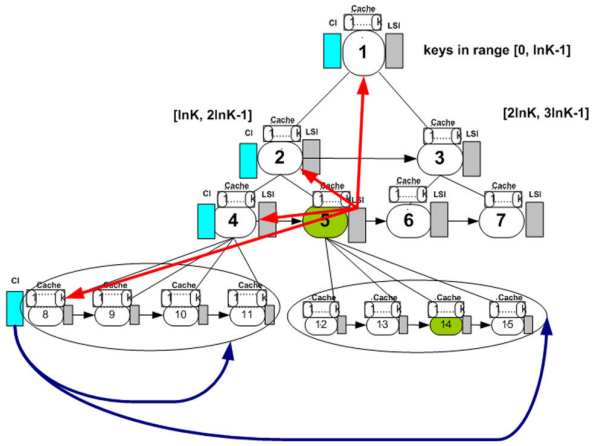

Figure 2: *The NBDT P2P System*

each key is distinct and as a result the probability of collisions is zero. Each key is stored at most in O(loglogN) levels. We also equip each peer with the table LSI (Left Spine Index). This table stores pointers to the peers of the left-most spine (for example in figure 2 the peers 1, 2, 4 and 8 are pointed by the LSI table of peer 5) and as a consequence its maximum length is O(loglogN). Furthermore, each peer of the left-most spine is equipped with the table CI (Collection Index). CI stores pointers to the collections of peers presented at the same level (see in figure 2 the CI table of peer 8). Peers having same father belong to the same collection. For example in the figure 2, peers 8, 9, 10, and 11 constitute a collection of peers. It's obvious that the maximum length of CI table is O$\sqrt{N}$. For example in figure 2 we are located at (green) node 5 and we are looking for a key k in [13lnn, 14lnn-1]. In other words we are looking for (green) node 14. As shown in [sioutas], the whole searching process requires O(loglogN)hops or lookup messages and that is also validated using the proposed simulator.

When we want to insert/delete a key/node from the structure, we initially search for the node that is responsible for it (using a number of O(loglogN) hops in worst-case) and then we simply insert/delete it from the appropriate node.

If new w'-bit keys, with w'>w, request to be inserted into the system, then we have to insert new peers on the network infrastructure and as a result we have to re-organize the whole p2p structure. In practice, such an expensive reorganization is very sparse. The new peers of NBDT are inserted at the end of the whole infrastructure consuming O(1) hops in worst-case. In particular, when a node receives a joining node request it has to forward the join request to the last node. The last node of NBDT infrastructure can be found in O(1) hops in worst-case by using the appropriate LSI and CI indexes.

If the load of some peer becomes zero, we mark as deleted

the aforementioned peer. If the number of marked peers is not constant any more then we have to re-organize the whole p2p structure. Based on the basic theorem of [kaporis 2003], if we generate the keys according to smooth distributions, which is a superset of regular, normal, uniform as well as of real world skew distributions like zipfian, binomial or power law (for details see [kaporis 2006]), we can assure with high probability that the load of each peer never exceeds polylogn size and never becomes zero. The latter means that with high probability split or delete operations will never occur. In other words, the re-organization of the whole P2P structure with high probability will never occur.

References

[Cox] Ingemar J. Cox, Matthew L. Miller and Jeffrey A. Bloom, Digital Watermarking. Morgan Kaufmann Publishers 2002.

[CSTB] Computer Science and Telecommunications Board, National Research Council. (1999). The Digital Dilemma: Intellectual Property in the Information Age (pp. 2-3). Washington: National Academy Press.

[House] House of Representatives. (1998, october). Digital Millennium Copyright Act.

[Randall] Randall Davis, "The Digital Dilemma", Communications of the ACM, Volume 44, February 2001, pp. 80.

[Wayner] P. Wayner, Disappearing Cryptography - Information Hiding: Steganography and Watermarking (Second, pp. 291-318). (2002). Morgan Kaufmann.

[Sioutas] S.Sioutas, "NBDT:An efficient P2P indexing scheme for Web Service Discovery", International Journal of Web Engineering and Technologies, Vol 4(1), pp. 95-113.

[Kaporis 2003] A. Kaporis et. al (2003) "Improved Bounds for Finger Search on a RAM", ESA, LNCS 2832, 325-336.

[Kaporis 2006] A. Kaporis et. al (2006) "Dynamic Interpolation Search Revisited", ICALP, LNCS 4051, 382-394.

[einhorn] M. Einhorn and B. Rosenblatt, (2005) "Peer-to-Peer Networking and Digital Rights Management - How Market Tools Can Solve Copyright Problems", Policy Analysis Journal, No. 534.

[cappellini] M. Barni, F. Bartolini, V. Cappellini, A. Piva, "A DCT-domain system for robust image watermarking", Signal Processing, "Special Issue on Watermarking", (66) 3 (1998), pp. 357-372.

4. Copyright

Part of the work has been developed in the framework of GSRT Praxitelis Project (Greece).

5. Conclusions

In this paper we focused on a P2P network infrastructure which allows broad digital content exchange while on the same time supports copyright protection and management through DRM technologies. In brief, a watermarking algorithm casts watermarking keys to the digital images and the same time the watermarking keys are being stored in the independed network Peers. Based in the NBDT system, in the steady state, in a N-node network, each node resolves all lookups via O(loglogN) messages to other nodes. Key updates require only O(loglogN) number of messages in worst-case. Node updates require O(1) number of messages in expected-case with high probability. The watermarking key detection process withih the P2P framework is very efficient and outperforms the most popular infrastructures used directly for many solutions for P2P information discovery. The key detection process is very important for the copyright owner because when succesful the copyright status of each digital image can be retrieved and evaluated. The future applicability of the proposed infrastructure is strong as it could be used for the creation of P2P environments, supported by GUIs, with which a user could exchange digital files while copyright protection occurs at the same time.