CERIAS Tech Report 2002-30

KEY COMMITMENT IN MULTIMEDIA WATERMARKING

by Radu Sion, Mikhail Atallah, Sunil Prabhakar

Center for Education and Research in
Information Assurance and Security,
Purdue University, West Lafayette, IN 47907

Key Commitment in Multimedia Watermarking (CERIAS TR 2002-30) [*]

Radu Sion, Mikhail Atallah, Sunil Prabhakar
Center for Education and Research in Information Assurance,
Computer Sciences, Purdue University
West Lafayette, IN, 47907, USA
http://www.cs.purdue.edu/homes/sion
[sion, mja, sunil]@cs.purdue.edu

October 23, 2002

Abstract

Many multimedia watermarking techniques [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] require the use of a secret key to detect/decode the watermark in/from the marked object. Court proofs of ownership are strongly related to the ability of the rights holder (i.e. Alice) to convince a judge (i.e. Jared) or a jury of the safety of the encoding/decoding key in the frame of the considered watermarking algorithm.

Multimedia Watermarking algorithms operate often in high bandwidth, noisy domains, that empower defendant (i.e. evil Mallory) court time claims of exhaustive key-space searches for matching keys. In other words, Mallory's position claims that Alice cannot prove her associated rights over the disputed content as the actual data domain in case allowed her to "try" different keys until one of them made the watermark magically "appear" in the (allegedly) un-marked object.

Watermarking algorithms in general and in the media framework in particular, would thus benefit from an intrinsic component of the security assessment step, namely a solution offering the ability to fight exactly such claims.

One mechanism for securing this ability is to precommit to the watermarking key, at any time *before* watermark embedding. Precommitting to secrets in the framework of watermarking presents a whole new set of challenges, derived from the particularities of the domain.

The main contribution of this paper is to define the main problem behind it and offer a solution to key precommitment in watermarking, solution augmented by a practical, illustrative example of an actual key precommitment method.

Given any watermarking scheme our solution increases its ability to "convince" that the associated watermark is not embedded through some post-facto matching key choice (or even fortuitously), and was in fact deliberately inserted.

In some sense we are providing a mechanism for the "amplification of convinceability" of any watermarking algorithm. That is, if the watermarked object makes it to court then its watermark proof is dramatically more convincing, and in particular immune to claims of matching key searches.

Thus, we introduce the main motivation behind precommitment to keys in the process of watermarking and present an algorithm for key precommitment, analyzing its integration as part of any existing watermarking application.

Our solution, while relying on new (e.g. tolerant hashing) and existing concepts (e.g. key-space size reduction, watermark randomization) ties them together to produce a drastic (i.e. to virtually 0) reduction of the probability of success in the case of random key-space searches for matching keys, thus making a convincing counter-point to claims as the one above.

We analyze trade-offs and present some alternative ideas for key precommitment. We discuss properties of the presented scheme as well as some other envisioned solutions.

1 Introduction

Digital Watermarking, in the traditional sense is the technique of embedding un-detectable (un-perceivable) hidden information into multimedia objects (i.e. images, audio, video, text) mainly to protect the data from unauthorized duplication and distribution by enabling provable rights over the content.

Proof of rights is usually achievable by demonstrating that the particular piece of data exhibits a certain rare property (read "hidden message" or "watermark"), usually known only to Alice (with the aid of a "secret"), the property being so rare that if one considers any other random piece of data, even

[*] Portions of this work were supported by Grant EIA-9903545 from the National Science Foundation, Contract N00014-02-1-0364 from the Office of Naval Research, and by sponsors of the Center for Education and Research in Information Assurance and Security.

similar to the one in question, this property is very improbable to apply. It is to be stressed here that the main focus in watermarking is on ’detection’ rather than ’extraction’.

Watermarking algorithms [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] often operate in high bandwidth, noisy, domains, that empower defendant (i.e. evil Mallory) court time claims of exhaustive key-space searches for matching keys. In other words, Mallory’s position claims that Alice cannot prove her associated rights over the disputed content as the actual data domain in case allowed her to “try” different keys until one of them made the watermark magically “appear” (see ”mark invertibility” in [14]) in the (allegedly) un-marked object. Watermarking algorithms in general and in the media framework in particular, would thus benefit from an intrinsic component of the security assessment step, namely a solution offering the ability to fight claims such as the ones above.

One mechanism for securing this ability is to pre-commit to the watermarking key, at any time *before* watermark embedding. Pre-committing to secrets in the framework of watermarking presents a whole new set of challenges, derived from the particularities of the domain.

The main contribution of this paper is to define, formalize and offer a theoretical solution to key pre-commitment in watermarking, augmented by a practical, illustrative example of an actual key pre-commitment method.

We introduce the main motivation behind pre-commitment to keys in the process of watermarking and present an algorithm for key pre-commitment, analyzing its integration as part of any existing watermarking application.

Given any watermarking scheme our solution increases its ability to “convince” that the associated watermark is not embedded through some post-facto matching key choice (or even fortuitously), and was in fact deliberately inserted at creation time.

In some sense we are providing a mechanism for the “amplification of convince-ability” of any watermarking algorithm. That is, if the watermarked object makes it to court then its watermark proof is dramatically more convincing, and in particular immune to claims of matching key searches as above.

As later argued, this comes at the expense (to an acceptable degree, we believe) of a slightly lower ability to resist generic attacks. Thus we provide a trade-off between “attackability” and “convince-ability”. A watermarking mechanism that features an intrinsic high level of overall resilience (e.g. many copies of the watermark embedded in the object) as is the case with high-bandwidth domains, can clearly benefit from such a tradeoff.

Thus, we introduce the main motivation behind pre-commitment to keys in the process of watermarking and present an algorithm for key pre-commitment, analyzing its integration as part of any existing watermarking application.

Our solution, while relying on new (e.g. tolerant hashing) and existing concepts (e.g. key-space size reduction, watermark randomization) ties them together to produce a drastic (i.e. to virtually 0) reduction of the probability of success in the case of random key-space searches for matching keys, thus making a convincing counterpoint to claims as the one above.

We analyze trade-offs and present some alternative ideas for key pre-commitment. We discuss properties of the presented scheme as well as some other envisioned solutions.

Thus, in this paper we explore how Alice convinces Jared the Judge in court, at watermark detection time, that the secret (watermarking key) associated with the detection process is not the result of an exhaustive search for matching patterns in the resulting object but rather was pre-committed to at watermark embedding time. By this, Alice fights Mallory’s claims that she might have exhaustively searched the key-space for a matching key that made the desired watermark “appear” out of the allegedly un-marked object. This is an essential step in increasing the security of many watermarking applications, where domain particularities allow for such exhaustive key searches yielding matching (key, watermark, object) tuples.

The paper is structured as follows. Section 2 defines the main motivation behind key commitment in watermarking and introduces its main associated challenges. Section 3 presents our solution while Section 4 discusses several improvements as well as a set of associated attacks. Section 5 outlines main conclusions and future envisioned developments.

2 Motivation and Challenges

Although some approaches aiming at asserting creation rights and time-stamping are based on publishing various types of hashes for every created digital object (i.e., possible target for watermarking) in a place beyond the owner’s control (e.g., in a dated newspaper), many other applications do not allow for this approach.

Market conditions and other considerations (e.g. desired stealthiness of the marking process, storage capacity required to store original objects and their associated information) often make the above scheme undesirable or cost-ineffective. Watermarking for rights-assessment becomes thus a competing approach for getting the job done.

Given applications functioning under the above assumption (i.e. impracticality of any type of public dated information source commitment/hashing) it becomes clear that any (pre)commitments to information associated with a watermarking application (e.g. secret key) also cannot be created by publishing but rather by making them part of the watermarking method. In order to pre-commit to

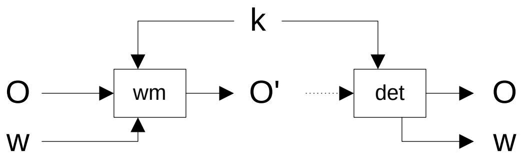

Figure 1: Symmetric 1-key watermarking (a) embedding [wm] and (b) detection [det]

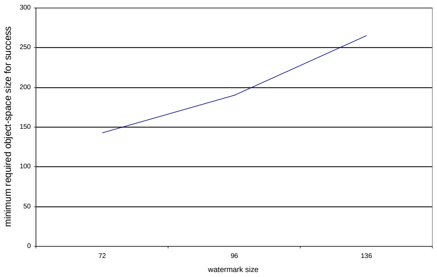

Figure 2: Average required minimum LSB space size for different watermark sizes in the case of experimental exhaustive searches for matching keys. An almost linear dependency can be observed.

a secret key, a special extension to existing watermarking algorithms needs to be defined that will effectively implement the pre-commitment step. The following subsections explore this and other aspects in more detail.

2.1 Scenario

A symmetric 1-key watermarking scheme is depicted in Figure 1. Watermark embedding (a) happens usually at object creation/release time whereas the detection process (b) is ultimately to happen in court [1], in the presence of Jared (i.e. the judge).

The ability to convince Jared or the jury of the association between the watermark and the given watermarked object (i.e., through the secret key, now to be revealed) is strongly related to proving that the given key is not the result of an exhaustive search of the key space (i.e. for the given object). In other words the given watermarking application has to be safe from claims that the data was "tortured until it confessed" (i.e. by exhaustive search), *after* creation/release time. This is of course particularly so if the watermark is short and where artificially lengthening it (e.g., by padding with 1s) is not practical (e.g., because of the object's inability to have long enough watermarks embedded in it).

Pre-committing to the secret key at embedding time is one solution to the above problem. The next subsection presents some of the challenges associated with key pre-commitment. In the following we introduce a simple illustrative example on how exhaustive search for a key in the key space can yield a certain desired watermark in the object space (e.g. LSB).

Note: Throughout the paper we are using as an application example a trivial watermarking method, namely LSB. As our contribution does not lie in the actual watermarking method but rather in the key pre-commitment algorithm, we decided to purposefully use a rather simple marking method as an illustrative deployment scenario, to direct the reader's focus on the *main* research presented by the paper, and not on the actual watermarking method used to illustrate it. Deployment in any other watermarking method is basically identical.

The theoretical grounding of LSB space key search is not trivial in the general case, as related research by Szpankowski [15] [2] and others [16], suggests, and is also outside the scope of this paper. Nevertheless we introduce a simple experiment as a proof of concept.

Experiment.

Given the class of LSB marking algorithms [1], a certain predefined/desired watermark w and digital images with LSB spaces (e.g. of size $n = 1024$ bits, an image of color depth 8 and file size over 8Kbytes etc), for illustrative purposes, we consider a trivial, text-book level, watermarking method M_{LSB} that simply selects a subset of the LSB space and replaces it with the watermark bits. The secret key k is then composed of the LSB-space indexes of the selected subset bits.

Then, given any random object and its associated LSB space, as above (e.g. 1024 bits, uniform distribution), there is a high probability of being able to find a key k that, if applied to o with the watermarking algorithm M_{LSB}, will yield (at detection time) the exact desired watermark (i.e. w == "copyright by smart people") In the following we compute this probability experimentally on a large set (10000) of generated, uniform distributed LSB spaces (bit strings) and then on 10 standard real world image LSB sets.

Given a certain length of the watermark, Figure 2 plots the average required random LSB space size for an exhaustive search approach to succeed in finding a matching key.

As expected, experiments confirmed the theoretical results. The required average minimum size of the LSB space for a success in finding a matching key in a exhaustive search

[1] We are aware of other instances when detection is deployed for various other reasons (e.g. online crawling and tracebacks etc) but in many applications, the ultimate main purpose of the watermark is to convince in court (or force settlement) once suspicions of illegal use surface.

[2] This research suggests, and partially computes and proves, the existence of a high probability associated with finding a given pattern (i.e. the desired watermark) as a subsequence within another larger string (e.g. the embedding space, LSB).

approach is only approximately *double* the bit-size of the considered watermark to be embedded.

Note: An easy observation to make here is that on average if the LSB space bit-size n is greater than $|w| \times 2^{|w|+1}$ then *any and all* watermarks of bit-size $|w|$ can be "found" by a key search.

For the considered real images the results are identical. This basically shows that finding a key that will match a desired watermark in any given image LSB space is extremely likely and probably easy.

This conclusion allows for court-time counter-attacks claiming that the watermarking key yielding the revealed watermark was actually searched for and not used at object-creation/watermark-embedding time. This issue is further explored by Craver et al in [14] (a.k.a. "mark invertibility").

In order to be suitably convincing in court, pre-commitment to the watermarking key is required at embedding time. The following subsection deals with challenges in designing a coherent key commitment scheme for watermarking.

2.2 Challenges

Associating both the mark and the required key pre-commitment with the resulting watermarked object presents an entirely new set of challenges. The impracticality to publish external help-information (e.g. hashes in dated newspapers), as outlined above, requires that (i) key-commitment can be directly derived from the object's content, producing a "self-contained" proof mechanism.

If the key space is dependent on the to-be-watermarked content, key selection becomes vulnerable to actual changes, even minor (e.g. attacks or even the watermarking process itself), of this very content. In other words, we would like to be able to produce the key-commitment information even after attacks and/or watermarking of the original content. Thus, (ii) there has to be a certain tolerance of the key pre-commitment scheme to minor changes of the original object (i.e. wrt. maximum allowable change in usability [17]).

Because of the "self-containment" required of the key-commitment information (i.e. within the watermarked object) another issue arises with respect to an exhaustive key-search attack that can be deployed on the distributed or published watermarked content. Thus, the key pre-commitment scheme has to either (iii) not make the key entirely derivable from the watermarked content or (iv) guarantee "enough" safety (i.e. in terms of computational complexity) that will make an exhaustive search attack infeasible. The following section presents a solution that takes into consideration the issues above.

3 Solution

Our solution starts by making the watermarking key-space directly associated (see requirement (i) above) with the to-be-watermarked object in a tolerant (see requirement (ii) above) and keyed/secret fashion (see requirement (iii) above). The actual association with the content as well as content change/attack immunity of the key-space is achieved through the novel concept of "tolerant hash". An exhaustive search for the watermark and the key is prevented (see requirement (iv) above) by means of encryption and key-space size self-tuning.

3.1 Tolerant hashes

Given a certain object to be watermarked, $o \in \mathbb{D}$, a watermarking algorithm, a distortion metric domain (i.e. see "usability domain" above, e.g. Human Visual System) with an associated distortion metric δ, and a maximum allowable distortion distance limit δ_{max}, we define a "tolerant hash" (with respect to the above given), as a function $J : \mathbb{D} \to \mathbb{Z}^+$ such that $\forall o' \in \mathbb{D}$ the following holds $J(o) = J(o') \iff \delta(o, o') < \delta_{max}$.

In plain wording, a *tolerant hash* is a function of a certain content which, while specific to the content, tolerates "minor" (i.e. in terms of the given distortion bounds determined by δ_{max}) changes to it [3]. The idea is to capture and quantify certain global specific properties of the content that are still preserved in the watermarking/attack process. Research by Ari Juels et. al. [18] investigates a related notion, suggestively qualified as "fuzzy commitment". While our tolerant hash concept presents idea similarities, it differs widely in its applicability as well as in the likely proposed implementations.

As an example, in our LSB scenario, if we consider the data objects in $\mathbb{D}$ as being represented as strings of bits, for the purpose of constructing a tolerant hash, we propose the metric δ to be the Hamming distance between considered objects.

Also, given the implicit framework assumption that modifications to the LSB space are not "noticeable", the maximum allowable distance δ_{max} is then naturally defined as the largest possible Hamming distance between two objects that differ only in their LSB space bits (it is trivial to show that this value is equal to the LSB space size).

The tolerant hash for the LSB example can be then defined as $J_{LSB}(o) = H(o_{no\_LSB})$ where o_{no_LSB} represents the object o with the LSB space removed (all the bits in o, without the LSB bits).

[3] Hence the name attribute "tolerant". This is a rebel hash that doesn't have traditional "good" hashing property and, in fact, does almost the exact opposite (i.e. tolerates minor changes to input). The term "hash" is used here to basically reflect the qualifications given to it by it's prefix, "tolerant" while still preserving the idea of a "summary of its input".

With respect to the given data and distortion metric domains, $J_{LSB}(\cdot)$ is a natural tolerant hash, by construction.

While domain specific solutions such as the one above, are probably more stable and better suited for each application, some other generic hash ideas are presented in Appendix B.

3.2 A preliminary solution

The first step involves randomizing (see challenge (iv) in Section 2.2) the watermark w before embedding such that even if a given attack round/method succeeds in detecting w, the attacker is not aware of it (otherwise the next attack step would have been a mark removal or destruction attempt) because it presents itself like white noise (see Figure 3).

We propose to randomize the watermark, i.e., make it look like white noise. This is achieved by keyed hashing step,

$$w = H(k_w, w_0, k_w) \tag{1}$$

where $H(\cdot)$ [4] is a one-way hash function (e.g. similar [5] to SHA, MD5), w_0 is the original watermark, k_w is a certain hashing key and w is the resulting watermark that will be used in the marking process later on. An alternative that does away with having to know w_0 before retrieving it is (if w_0 has the right length)

$$w = E_{k_w}(w_0) \tag{2}$$

or, if w_0 is too short,

$$w = E_{k_w}(k_w, w_0) \tag{3}$$

where $E_{k_w}(\cdot)$ can be any cryptographic encryption primitive which encrypts its input with the key k_w.

Note: The key k_w is required (e.g., vs. the simple case of $w = H(w_0)$) in order to prevent semantic attacks in which Mallory is aware of the actual watermark text (w_0) or the nature of the text (and based on that he can drastically limit the space of an exhaustive search attack, obtaining a valid w).

The second step reduces the key-space size (of the watermark insertion key) by a procedure ensuring that a successful exhaustive search for a key that matches a desired watermark is highly unlikely (see LSB scenario above).

Let $m \in \mathbb{Z}^+$, an integer to be made public and part of the overall watermarking procedure description. Let $c \in [0, m]$ be a secret integer, randomly selected [6] at mark embedding time. Let there be a certain domain specific tolerant hash $J : \mathbb{D} \to \mathbb{Z}^+$ (e.g. J_{LSB} as above). Then the watermark embedding key $k \in \mathbb{K}$ to be used in the marking algorithm is defined by

$$k = H(c, J(o), c) \tag{4}$$

where $H(\cdot)$ is a one-way hash function (e.g. SHA), and o is the to-be-watermarked object.

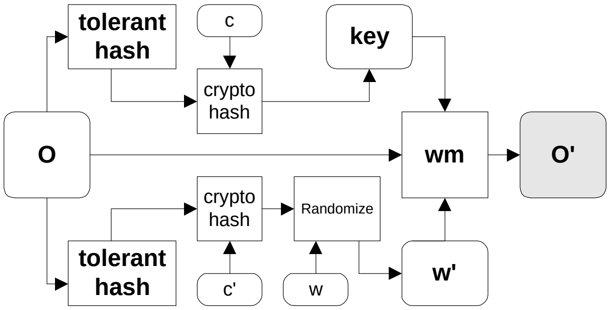

Figure 3: Through pre-commitment both the encoding key and the watermark are committed to *before* watermarking.

It is assumed here that the resulting key will allow the watermarking algorithm to embed the desired mark. If this is not the case and the given key simply won't work (i.e. within given object distortion bounds), selecting a different c yields a completely different key.

Note: An interesting domain-specific issue can be raised in the case that none of the allowed c's will result in a usable key. First solution ideas include selecting a different tolerant hash, selecting a different one-way hash-function, etc.

A special property of this scheme is the fact that it is self-contained (i.e. it only depends on the considered data domain and the to-be-watermarked object size) and can be safely published (i.e. m can be made public) as a specification of the overall watermarking method.

3.3 Benefits

In the following we discuss some of the benefits induced by the pre-commitment method as described above.

(a) In the traditional, non-pre-committed approach, given the LSB space size n and the considered watermark w, the probability that a given randomly selected embedding key $k_{rand} \in \mathbb{K}$ matches the given watermark [7] can be computed as follows:

$$P_{sucess} = \frac{avg\_rnd\_wm\_occ}{key\_space\_size} \tag{5}$$

[4] Remember that many well-known cryptographic hashes (e.g. MD5, SHA) suffer from a construction weakness allowing for input append if not keyed properly. This is why we parameterize $H(k_w, w_0, k_w)$ instead of $H(k_w, w_0)$. Also in this notation, "k_w, w_0" denotes the concatenation of the bit string k_w with the bitstring w_0.

[5] A simple transform can be applied on the bit-output of SHA/MD5 in order to yield an appropriate number of result bits. For example XOR-ing all words in the SHA/MD5.

[6] See 4 for an extension to this idea.

[7] We naturally use this probability as a metric of success for an exhaustive search approach.

where $avg\_rnd\_wm\_occ$ is the experimentally or theoretically determined number of expected random occurrences of a given watermark, in a given data space, and key_space_size is the size of the embedding key space considered (e.g. in our case $C_n^{|w|}$).

Once pre-commitment is introduced, as a direct result of key-space reduction (to a cardinality of m), the probability becomes:

$$P'_{sucess} = \frac{m}{C_n^{|w|}} \times P_{sucess} \qquad (6)$$

This is due to the fact that the considered key space size was "compressed" in equation 4. Associated, also the probability of success in a random search for a matching key decreases similarly.

If the LSB space size is $n = 100000$ (e.g. aprox. 99KBytes image file), the considered watermark is of size 100 bits, and we use $m = 100000$ this results in a very large probability reduction ratio of $\frac{100000}{C_{100000}^{100}}$. In other words, if the original key space allowed a random search with a probability of success for each step being P_{step}, by using pre-commitment as above this probability is reduced to

$$P'_{step} = P_{step} \times \frac{100000}{C_{100000}^{100}} \qquad (7)$$

Also, binding and constraining (i.e. in (4), by the one-wayness of the hash) the key-space lowers dramatically (cryptographic hash inverse computation complexity) the probability that a matching key, (exhaustively searched for) will satisfy (4) and that also a corresponding c exists and is found.

Thus, the watermarking key is recoverable in court directly from the watermarked content in such a way as to also prove commitment. Commitment is guaranteed partly by the one-wayness of the considered hash functions and partly also by the key construction mechanism which is deterministic and derives the key from the content.

(b) While exhibiting all these benefits, reducing the key space size also exhibits some apparent drawbacks. One main concern is for an adversary (e.g. Mallory) not to be able to gain relevant information about the watermarking key from knowing $[0, m]$ (remember m is public). The assurance here is brought about by the initial watermark randomizing step. Its security lies in the inherent accepted non-invertibility of cryptographic hashes.

There are many watermarking attacks that can be considered with respect to any watermarking method. It is out of the scope of this paper to analyze these attacks; we rather assess any possibly induced weakness associated with the newly introduced pre-commitment step. By reducing the key space size we apparently made it easier for the attacker to exhaustively deduce/infer the embedding key.

More specifically, Mallory has an option of performing an exhaustive search on the key space, of size m. But, even given knowledge about the original watermark (i.e. he knows w_0 and/or is able to identify it if seen) he cannot properly detect it because of the randomization step (1).

Because Mallory also operates within tolerable distortion bounds (i.e. trying to maintain "value", see Appendix), an implicit assumption is made that he can not afford to simply randomly pick keys and "remove" marks (by altering the content) because the object will almost surely be damaged beyond acceptable distortion limits [17].

The only viable attack decision with a potentially non-null success probability is choosing a random $c' \in [0, m]$ and assuming that it indeed is the initially chosen c in the embedding process. The probability of success of this approach as a whole is $\frac{1}{m}$, in the considered case $\frac{1}{100000}$. We would like to outline here again the fact that Mallory has only one chance at this and cannot "try" it several times as the required data changes will rapidly degrade the value of the already watermarked object, depending on the actual underlying watermarking algorithm.

Nevertheless, this is where the main trade-off of our solution becomes clear. The ability to fight exhaustive key-search claims comes at the expense (to an acceptable degree, we believe) of a slightly lower ability to resist single generic attacks. Thus our solution provides a trade-off between "attackability" and "convince-ability". A watermarking mechanism that features an intrinsic high level of overall resilience (e.g. many copies of the watermark embedded in the object) as is the case with high-bandwidth domains, can clearly benefit from such a tradeoff.

4 Critique

In this section we are introducing certain improvements as well as a discussion on several interesting attack approaches to the provided algorithm.

4.1 Improvements

The above scheme can be improved in ways that we explore below.

4.1.1 Trial-and-error on k_w

One weakness resides in the randomizing step (1), namely in the fact that the space of the resulting watermark w can be subjected to a similar (although much harder, based on the one-wayness of the hash) trial-and-error attack, by choosing different values for k_w, resulting in different values for w.

In other words, we apparently are back to square one, except that the trial and error problem has shifted from the watermark insertion key to the key used for randomizing. The claim of the adversary (i.e., Mallory) in court can be that Alice first "guessed" a value for c, generated an associated watermarking key k with (4) and then retrieved a watermark w (meaningless) out of o'. Using this watermark Alice then could have tried different values for k_w in order to "reverse-engineer" (1) until satisfied for a desired w_0 (e.g., saying "Copyrighted by Alice").

Although improbable (i.e. much harder, based on the one-wayness of the hash), the above claim needs to be properly addressed.

One simple solution to this issue is to bind (i.e. again, in a one-way sense) c to k_w and w_0. In other words instead of choosing a random c in step (4) let c be

$$c = H(k_w) \bmod n \tag{8}$$

In this case, any "reverse-engineered" w_0 derived out of (1) will also have to satisfy (8), with respect to the supposedly malevolently chosen c, which does not allow room for trial-and-error attacks.

Another solution is to simply apply our key space reduction method also on k_w. More formally, let there be $m' \in \mathbb{Z}^+$, an integer to be also made public and part of the overall watermarking procedure description. Let $c' \in [0, m']$ be a secret integer, randomly selected at mark embedding time. Let there be a certain domain specific tolerant hash $J : \mathbb{D} \to \mathbb{Z}^+$ (e.g. J_{LSB} as above). Then the watermark randomizing key $k_w \in \mathbb{K}$ is defined by

$$k_w = H(c', J(o), c') \tag{9}$$

where $H(\cdot)$ is a one-way hash function, and o is the to-be-watermarked object. The security discussion applies as above, a successful whitening key detection attack having a success probability of $\frac{1}{m'}$. Keep in mind that this is to be considered additionally to the required next attack step (corresponding to the actual pre-commitment), namely detection/removal of the actual watermark w.

Note: This second solution effectively transforms the watermarking key space from $\mathbb{K}$ to $\mathbb{Z}^+ \times \mathbb{Z}^+$, as c and c' become the actual sole secret keys for the algorithm.

4.1.2 A randomized but unchanging w

If the same w_0 and k_w are used to watermark many objects, then (1) implies that the same w will be used in all of them. By trying different c values for each such object and observing the resulting extracted watermark (if access is available to a set of watermarked objects and associated marks), the adversary can figure out w and also which c was used in each object ! To achieve a variability in w even when the same w_0 and k_w are used for many objects, one could introduce $J(o)$ inside (1), e.g.

$$w = H(k_w, w_0, J(o), k_w) \tag{10}$$

which would prevent the above-mentioned attack.

4.1.3 Variants

The actual commitment to the key in our presented solution is provided indirectly by the point outlined in (a) above. Nevertheless we envision different pre-commitment schemes where the commitment is made more explicit.

One idea would be to make a certain key pre-commitment (e.g. hash of key) part of the actual watermark in itself. On the other hand, this immediately poses issues related to collusion-safety because different watermarks (with different keys and associated commitments) will be used for different distributions/publications of the same object.

4.2 Discussion

Key pre-commitment addresses the issue of court-convince-ability in the framework of watermarking for copyright protection. It can be used as an effective tool in fighting Mallory's claims that Alice exhaustively searched the key-space for a matching key, given an associated watermark.

Key pre-commitment does *not* address other generic information hiding and watermarking problems. One example of such a problem is the scenario of multiple watermark embeddings within the same object. This case has the potential to lead to a court conflict as to which watermark is the "authentic" (i.e. original) one etc. If the underlying watermarking method allows for multiple mark embeddings then a solution to this issue has to be found as part of that method.

In the following we are going to address some of the issues and apparent problems that we feel still need discussion.

4.2.1 Searching for both watermark/key

One could argue that after all, reducing the key space size does not help too much if the space of the watermark to be embedded is of infinite size. An exhaustive search in the (key,watermark) space could still be very successful given the large cardinality of this space (induced by the watermark space).

One approach to the above problem starts from the observation that in real scenarios, usually the space of possible watermarks is of finite size, often much much smaller than the cardinality of the initial key space (e.g. what is the number of meaningful phrases in English composed of a total of 20 letters).

Moreover, a requirement (e.g. in court) can be easily enforced, stipulating that the actual watermarks need to conform to a special format, thus reducing the cardinality of the watermarks-space. For example mandating a format of "(C) by CompanyName" as the only accepted watermark for a given company reduces the effective mark-space to one. Also, in the present case, the final embedded watermark itself is pre-committed to as required in equation 10, making the search practically impossible, bound by the cryptographic non-invertibility of the one-way hash.

4.2.2 Replacing the watermark

In this scenario, an attacker (e.g. somebody wishing to claim rights over the object, post-creation, post-watermarking) chooses parameters leading to an arbitrary key k, then slightly modifies the object, but keeps the key k unchanged (as k depends only on the "significant" parts of the object). Through successive "slight" modifications to the object, the attacker hopes to find again a detectable watermark, different from the initial one.

This scenario is yet another example of a generic watermarking-related problem which is not intended to be addressed by key pre-commitment. Modifying the initial watermarked object such as to remove or alter the watermark is a traditional attack in watermarking [1] [14] [8] and is to be dealt with by the particular watermarking method in question. Nevertheless, in the following we are trying to provide a brief insight into how this is to be addressed.

The main power of watermarking [17] lies in the exact dilemma the attacker faces by not knowing "where" the watermarked object is positioned (i.e. in the "usability domain") with respect to the original. In other words, the attacker does not possess any information as to what kind of modifications can still be performed to the object without destroying it's value. The object could be at the boundary of the usability domain, as proposed in [17] and "playing" with it could very likely result in an invalid object (i.e. distorted, without value).

4.2.3 Fixing m

In this case, an attacker could "play" with values for m until one is found that satisfies, e.g. provides a large enough key-space for a matching key to be found by exhaustive search. Thus Mallory would claim that Alice (i.e. the potential attacker in this scenario) did exhaustively search in both the space of all possible m values and the space associated keys to find a matching key.

To address this issue one has to take into account the fact that while there is definitely room for a choice of m, it is nevertheless an *algorithm specific* parameter. A specific party (e.g. Alice) cannot afford using a different m every time the algorithm is applied, or the result of the method will not not yield the promised court convince-ability. Rather, once a certain m value is chosen (or a very small finite set of values) it has to be used throughout for every and all application of the algorithm.

Thus, to fight Mallory's claim, Alice would only need to provide proof that the same m value is used consistently to mark Alice's works.

5 Conclusions

We discussed key pre-commitment within the frame of multimedia watermarking, and provided an algorithm that can be applied to existing watermarking schemes, enhancing their ability to convince in court by fighting claims of exhaustive key-space searches for a desired watermark matching key.

The solution we presented is based on traditional concepts as well as new ideas, such as tolerant hashing and key-space reduction. We discussed associated trade-offs and presented some alternative ideas for key pre-commitment schemes.

Future efforts should explore domain specific applications and ways of better empowering them in court through the use of key pre-commitment schemes.

The relationship between watermarking with key pre-commitment and the redundant use of short watemarks needs also to be investigated. We envision that key pre-commitment will allow for very short watermarks, which in turn makes higher watermark resilience possible.

Acknowledgements

We thank Professor Wojciech Szpankowski for his suggestions of relevant literature regarding subsequence occurrence analysis. We also would like to thank our anonymous reviewers who guided our job of clarifying several issues in the original algorithm description.

Appendix A: Watermarking Model

An extended theoretical model for watermarking is out of the scope of this paper. Initial steps can be found in [17] as well as in [19] [20] [21] [22] [23] related research in the broader area of steganography and information hiding. Nevertheless, various comments and suggestions led us to believe it might be a good idea to include a short model introduction in order to make the present paper more self-contained. Thus we are including a short summary of the main considered formalism for watermarking, as presented more in depth in [17].

One fundamental difference between watermarking (i.e. mainly for rights protection) and steganography in general resides exactly in the main applicability and descriptions of the two domains. Steganography's main concern lies in Alice and Bob being able to exchange messages [24] [25] [26] in a manner as resilient and hidden as possible, through a medium controlled by malicious Wendy. On the other hand, digital watermarking is usually deployed by Alice to prove ownership or any other right over a piece of data, to Jared the Judge, usually in the case when Tim Mallory, the Thief benefits from using/selling that very same piece of data or maliciously modified versions of it.

Proof of rights is usually achievable by demonstrating that the particular piece of data exhibits a certain rare property (read "hidden message" or "watermark"), usually known only to Alice (with the aid of a "secret"), the property being so rare that if one considers any other random piece of data, even similar to the one in question, this property is very improbable to apply. It is to be stressed here that the main focus in watermarking is on 'detection' rather than 'extraction'.

Let $\mathbb{D}$ be the domain of all possible data objects to be considered for watermarking (e.g. digital images). Objects $o \in \mathbb{D}$ have associated value induced by the object creator. Watermarking tries to protect this association between the value carrying object and its creator.

Usability Domain: Complex objects can exhibit different value levels when put to different uses. We need a way to express the different associated values of objects, in different *usability domains*. Intuitively, a *usability domain* models different "uses" a certain object might be subjected to.

Usability: *Usability* is a measure of how "useful" an object can be with respect to a given domain. The concept of usability enables the definition of a certain threshold below which the object is not "usable" anymore in the given domain. In other words, it "lost its value" to an unacceptable degree. The notion of usability is related to *distortion*. A highly distorted object (e.g. as result of watermark embedding or attacks) will likely suffer a drop in its distortion domain usability.

Usability Vicinity: The *usability vicinity* of an to-be-watermarked object defines a set of objects that are not to far away (i.e. still acceptable, usable) from a given reference object. The *radius* of the vicinity is defined by the distance to the reference object of the "farthest" object within the vicinity.

Note that the usability vicinity of a certain object $d \in \mathbb{D}$ with respect to a considered set of usability domains $V \subset \mathbb{U}$ defines actually the set of possible watermarked versions of d with respect to V and Δu_{max}.

Watermark: A *watermark* can be defined as a special induced (through watermarking) property of a certain watermarked object $o' \in \mathbb{D}$, so rare, that if we consider any other object $x \in \mathbb{D}$, with a "close-enough" usability level to the original object o, the probability that x exhibits the same property can be upper-bound [8].

Watermark Power: The *power* of a certain watermark is directly related to its convince-ability towards Jared the Judge. The weaker the watermark (higher the false hit probability upper bound) the less convincing it will be. To be noted that this definition is not necessarily linked to the traditional bandwidth assessment approaches as it entails considering a multitude of other factors, such as attacks.

Attack: An *attack* simply tries to maintain the attacked watermarked object within the permissible usability vicinity of the original non-watermarked one, while making it impossible to recover the watermark.

Main Watermarking Challenge: Power and Usability. The main challenge of watermarking lies with keeping the resulting marked object within a certain permissible usability vicinity of the original while maximizing a certain metric (linked usually to mark resilience in the considered value domain, e.g. Human Sensory System, and/or attacks) related to persuasion ability in court.

Appendix B: Examples of Tolerant Hashes

(a) A characteristic of the content that can be partly used in defining a hash as above, is the "compressibility" of the given content. That is, given a certain compression algorithm (e.g. Huffman), what is the maximum compression ratio we can get after a pre-determined number of rounds.

Note: In trying to capture something specific, associated to a certain given content, but also resilient enough so that minor changes in the content will not change it, we encountered the proven idea of mapping the content data into a new domain and trying to find some properties of the mapping result that satisfy the original requirements.

(b) One simple mapping brings a one-dimensional data into a multiple dimensional space. For example it is possible to map the data to a 2 dimensional function defined by the following: starting in the origin of the coordinate system, if the next encountered bit is 1 advance 1 position on the oX scale and go "up", otherwise advance and go "down". The overall function shape can be integrated and the result is empirically proven to be quite content specific.

(c) Another mapping to a two dimensional space can be defined by simply considering each pair of content bytes as a (x, y) "point" coordinate. After plotting all the "points", it is proven that a fairly resilient property of the numbers involved in defining the plot is determined by repeatedly "peeling-off"

[8] A slightly different, more appropriate notation will be used throughout the paper, with clarifications where necessary. We considered it to be benefice for us to remain faithful to our original paper [17] in this short summary.

the convex hull until no more points remain on the plot [27]. The number of times we were able to perform the peel off as well as the series defined by the number of points peeled off in each round is proven to be very content specific and intuitively quite change tolerant.

(d) Of much success in the image watermarking community, are transforms like the DCT (deployed mainly in JPEG watermarking) that map content into the frequency domain. The important transform coefficients in the new domain are then used for storing watermarks by various altering methods.

Whereas we could certainly use the same transform in the case of a known JPEG image content, by assuming generality this is certainly not possible in the given form.

But still the idea is very relevant to the case. The fact that minor changes in the DCT coefficients (in the transform domain) lead to minor, mostly un-noticeable (wrt. the considered usability domain being defined by the Human Sensory System) changes in the result (i.e. back in the image domain) as well as the fact that DCT coefficients are quite content specific, lead to the idea that maybe using the inverse procedure will yield the desired results.

That is, we estimate that minor content changes will have little effect on corresponding transform coefficients. Thus, given a certain one-dimensional content bit-string, the corresponding hash value will be composed of a weighted combination of the significant transform coefficients.

Whereas using a transform in computing tolerant hashes can be used for various content, it does not make use of any particularities of specific types of content. For example if the node content is an JPEG image (e.g. relational multimedia database), a generic transform applied to an one-dimensional data view might be sub-optimal in that it wouldn't capture image features which, if captured, would certainly increase the level of specificity and graceful degradation with minor changes. In that case, using feature extraction algorithms (e.g. property histograms) and/or DCT transforms will certainly yield better results.

In the case of natural language (NL) content [28], capturing much of the specifics can be done by translating syntax trees and semantic relationships into certain characteristic values (e.g. by using Planted Plane Cubic Tree [29]).

References

[1] M. Atallah, V. Raskin, C. F. Hempelmann, M. Karahan, R. Sion, and K. E. Triezenberg. Natural language watermarking and tamperproofing. In *Lecture Notes in Computer Science, Proc. 5th International Information Hiding Workshop 2002*. Springer Verlag, 2002.

[2] W. Bender, D. Gruhl, N. Morimoto, and A. Lu. Techniques for data hiding. *IBM Systems Journal*, 35(3–4):313–336, 1996.

[3] B. Chazelle. On the convex layers of a planar set. *IEEE Transactions on Information Theory*, 31:509–517, 1985.

[4] G. Cohen, S. Encheva, and G. Zemor. Copyright protection for digital data. In *Workshop on Watermarking and Copyright Enforcement*, Paris France, 1999.

[5] I. Cox, J. Bloom, and M. Miller. Digital watermarking. In *Digital Watermarking*. Morgan Kaufmann, 2001.

[6] I. J. Cox and J.-P. M. G. Linnartz. Public watermarks and resistance to tampering. In *International Conference on Image Processing (ICIP'97)*, Santa Barbara, California, U.S.A., 26–29 Oct. 1997. IEEE.

[7] I. J. Cox, M. L. Miller, and A. L. McKellips. Watermarking as communications with side information. *Proceedings of the IEEE (USA)*, 87(7):1127–1141, July 1999.

[8] S. Craver. On public-key steganography in the presence of an active warden. In *Information Hiding*, pages 355–368, 1998.

[9] S. Craver, N. Memon, B.-L. Yeo, and M. M. Yeung. Resolving rightful ownerships with invisible watermarking techniques: Limitations, attacks, and implications. *IEEE Journal of Selected Areas in Communications*, 16(4):573–586, 1998.

[10] E. J. Delp. Watermarking: Who cares? does it work? In J. Dittmann, P. Wohlmacher, P. Horster, and R. Steinmetz, editors, *Multimedia and Security – Workshop at ACM Multimedia'98*, volume 41 of *GMD Report*, pages 123–137, Bristol, United Kingdom, Sept. 1998. ACM, GMD – Forschungszentrum Informationstechnik GmbH, Darmstadt, Germany.

[11] S. K. (editor) and F. P. (editor). Information hiding techniques for steganography and digital watermarking. In *Information Hiding Techniques for Steganography and Digital Watermarking*. Artech House, 2001.

[12] B. Ellis. Public policy: New on-line surveys: Digital watermarking. *Computer Graphics*, 33(1):39–39, Feb. 1999.

[13] P. Flajolet, Y. Guivarc'h, W. Szpankowski, and B. Vallee. Hidden pattern statistics. In *Automata, Languages and Programming*, pages 152–165, 2001.

[14] A. Juels and M. Wattenberg. A fuzzy commitment scheme. In *ACM Conference on Computer and Communications Security*, pages 28–36, 1999.

[15] M. Kobayashi. Digital watermarking: Historical roots. IBM Research Report RT0199, IBM Japan, Tokyo, Japan, Apr. 1997.

[16] M. Kutter and F. A. P. Petitcolas. A fair benchmark for image watermarking systems. In *citeseer.nj.nec.com/article/kutter99fair.html*, pages 226–239.

[17] P. Moulin, M. K. Mihcak, and G.-I. A. Lin. An information–theoretic model for image watermarking and data hiding. In *(manuscript)*, 2000.

[18] P. Moulin and J. O'Sullivan. Information-theoretic analysis of information hiding. In *P. Moulin and J. A. O'Sullivan, Information-Theoretic Analysis of Information Hiding*, 1999.

[19] J. Palsberg, S. Krishnaswamy, M. Kwon, D. Ma, Q. Shao, and Y. Zhang. Experience with software watermarking. In *Proceedings of ACSAC, 16th Annual Computer Security Applications Conference*, pages 308–316, 2000.

[20] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn. Attacks on copyright marking systems. In D. Aucsmith, editor, *Information Hiding: Second International Workshop*, volume 1525 of *Lecture Notes in Computer Science*, pages 218–238, Portland, Oregon, U.S.A., 1998. Springer-Verlag, Berlin, Germany.

[21] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn. Information hiding - a survey. *Proceedings of the IEEE*, 87(7):1062–1078, July 1999. Special issue on protection of multimedia content.

[22] B. Pfitzmann. Information hiding terminology. In R. Anderson, editor, *Information hiding: first international workshop, Cambridge, U.K., May 30–June 1, 1996: proceedings*, volume 1174 of *Lecture Notes in Computer Science*, pages 347–350, Berlin, Germany / Heidelberg, Germany / London, UK / etc., 1996. Springer-Verlag.

[23] M. Rgnier and W. Szpankowski. On pattern frequency occurrences in a markovian sequence? *Algorithmica (22)*, 1998.

[24] C. Shannon. Communication theory of secrecy systems. *Bell System Technical Journal*, 28:656–715, 1949.

[25] R. Sion, M. Atallah, and S. Prabhakar. Power: Metrics for evaluating watermarking algorithms. In *Proceedings of IEEE ITCC02, CERIAS TR 2001-55*. IEEE Computer Society Press, 2002.

[26] M. D. Swanson, B. Zhu, and A. H. Tewfik. Audio watermarking and data embedding – current state of the art, challenges and future directions. In J. Dittmann, P. Wohlmacher, P. Horster, and R. Steinmetz, editors, *Multimedia and Security – Workshop at ACM Multimedia'98*, volume 41 of *GMD Report*, Bristol, United Kingdom, Sept. 1998. ACM, GMD – Forschungszentrum Informationstechnik GmbH, Darmstadt, Germany.

[27] G. Voyatzis, N. Nikolaidis, and I. Pitas. Digital watermarking: an overview. In S. Theodoridis et al., editors, *Signal processing IX, theories and applications: proceedings of Eusipco-98, Ninth European Signal Processing Conference, Rhodes, Greece, 8–11 September 1998*, pages 9–12, Patras, Greece, 1998. Typorama Editions.

[28] J. Zhao and E. Koch. A generic digital watermarking model. *Computers and Graphics*, 22(4):397–403, Aug. 1998.

[29] J. Zhao, E. Koch, J. O'Ruanaidh, and M. M. Yeung. Digital watermarking: what will it do for me? and what it won't! In ACM, editor, *SIGGRAPH 99. Proceedings of the 1999 SIGGRAPH annual conference: Conference abstracts and applications*, Computer Graphics, pages 153–155, New York, NY 10036, USA, 1999. ACM Press.