Key issues for watermarking digital images

D. Augot and C. Fontaine

INRIA Domaine de Voluceau, B.P. 105 F-78153 Le Chesnay Cedex,FRANCE

ABSTRACT

This paper discusses secure architecture and protocols for managing Intellectual Property Rights in distributed content databases in a close environment. This discussion has been conducted within the European project AQUARELLE.

This paper presents a short survey of watermarking technologies and focuses on functionalities offered by such techniques. We propose the terms of watermarking, fingerprinting and monitoring. For our implementation, we have worked with the Université catholique de Louvain (UCL). This work is joint work with Jean-Francois Delaigle.

Next we focus mainly on keys issues, and conclude that a trusted third party is needed to establish a verification service of watermarks. Next the DHWM key exchange is presented, based on the simple idea that watermarking and verification can be separated. This scheme uses the Diffie-Hellman key-exchange protocol. Next some hints on the implementation of the scheme and on its correctness are given.

Keywords: IPR protection, watermarking, Key exchange, Aquarelle, multimedia distributed system.

1. INTRODUCTION

The new techniques of watermarking (or fingerprinting or stamping) digital images consists in hiding an invisible and robust mark into an image. This information should be sufficient to identify the copyright owner of the image: watermarked images can be traced to find their originator or their owner. The current technology does not enable copyright owners to protect their images, and many services are blocked in their development.

Watermarking, or *embedding* is a very new and complex technology, which hardly seems by now scalable to the whole Internet. Indeed, the technology is not as strong as classical cryptology, and reasonable attacks can be attempted with success.

This paper describes the solution devised for a closed environment designed for the access of the European Cultural Database. This work is done for the Aquarelle European Project. In this project, multimedia data is more clearly defined, and users are also well known and identified. In that context, a reasonable solution can be tailored.

A trusted third party TTP is introduced. Its role is to check the watermarked images. A first scheme is presented, and our scheme, named DHWM, improves that scheme by making use of the Diffie-Hellman protocol.

The paper is structured as follows: the second section briefly surveys techniques relative to watermarking, and fixes some terminology. Third section discusses key issues, giving user requirements and technical constraints. In section four, the algorithm from UCL is sketched. Section five introduces the DHWM functional model, using the Diffie-Hellman protocol.

2. A SHORT SURVEY AND A TERMINOLOGY

The following terms may be encountered “marking”, “fingerprinting”, “data hiding”, “steganography”, “label embeddings”, “watermarking” etc. We set a terminology and briefly survey the functionalities claimed by different propositions, since the objectives are quite different depending on the authors. We will not discuss the technicals properties of the algorithms (robustness, invisibility, speed ...) but only their aims and objectives.

Other author information: (Send correspondence to D. Augot.)
D. Augot.: E-mail: Daniel.Augot@inria.fr
C. Fontaine.: E-mail:Caroline.Fontaine@inria.fr

2.1. Classical cryptography

The following cryptographic techniques are not able to prevent fraudulent use of the images:

encryption encryption protects the images during their transmission. With encryption, an eavesdropper does not have access to the on-line image when it is transfered. But when the user has deciphered the image, then this image does not have any copyright protection anymore.

signature the owner of the image may electronically sign the image (with a hash function and a signature algorithm), but since the signature is added as a suffix to the image, it can easily be removed by anyone who gets the image.

2.2. Terminology

Following our collegues at UCL, we distinguish the following types of watermarking scopes.

steganography is a very generic concept that consists in hiding messages in a way that eavesdroppers or any monitors do not even know that there is a communication and a message is being sent. Many data hiding techniques were invented for this purpose. Those techniques inspired the development of watermarking algorithms for copyright protection.[9]

watermarking is the robust embedding of a copyright information (e.g. time and date, copyright identifiers or simply a correlation pattern) into a content. This content may be a text[12,1] or audio content[16] but most of the time watermarking is applied to still or moving images. This paper is focusing on images watermarking for both still pictures and motion pictures applications. In the current state-of-the-art, watermarking uses symmetric keys, in the sense that a secret key is used to hide data in a robust way and the same key is used to retrieve the data.

fingerprinting consists in uniquely marking and registering each copy of the data. This marking allows a distributor to detect any unauthorized copy and trace it back to the user. Fingerprinting englobes most often data hiding techniques and cryptographic protocols. Fingerprints have to resist to collusions attacks. It must be very difficult for a set of users to collaborate together and alter fingerprints by merging their copies. Data hiding techniques for fingerprinting can be for instance watermarking techniques but data can also be physically hidden in the media that support the data.[14]

2.3. Watermarking technologies

In this section, we make a short overview of most popular watermarking methods from different universities and companies. In the following "survey", we describe roughly the methods, but we mainly focus on the functional aspects of the methods. Section 4 will describe the watermarking algorithm that has been chosen in the Aquarelle project.

- G. W. Braudaway et al.[4] introduce a "visible watermark". It clearly identifies the ownership, and allows all image details to be seen through it. It is robust enough such that any attempt to remove it alters the image. However, the main drawback is of course that it reduces the quality of the picture.
- The algorithm introduced by R.G. van Schyndelin et al.[15] is envisaged to have application in image tagging, copyright enforcement, counterfeit access and controlled access, although the authors do not explain how to use their algorithm to perform these functionalities.
- J. Brassil et al.[1] apply electronic marking to textual document, by word or line shifting. An indiscernible codeword is added to the document, and it identifies the registered user to whom the document has been delivered. This can be applied on non-ascii text representation, and not to images.
- E. Koch et al.[10,17] define the following requirements for an invisible copyright label
  1. The image must contain a label or code, which marks it as the property of the copyright holder.
  2. The image data must contain a user code, which verifies that the user is in legal possession of the data.

3. The image data is labeled in a manner which allows its distribution to be tracked. Unfortunately, the invisibility of the watermark is not totally guaranteed. The watermark is embedded in chosen DCT coefficients of 8x8 blocks. In order to be resistant against compression, the chosen DCT coefficients have to be quantified, which means that marked blocks are altered. This feature can be damageable for high quality pictures such as museum images.

- I. J. Cox et al.[5] consider watermarking using spread spectrum.[8] Their mark identifies ownership and the user who got the image. This method has good robustness properties, however, both the original image and the marked one are needed to check the mark.
- F.M. Boland et al.[2] introduce an algorithm for invisibly marking an image. Here again, it is needed to have both the original image and the watermarked one to check the mark.
- N. Morimo et al.[16] focus on the notion of data hiding. They envisage the application to the problem of copyright proving and to the content integrity, but the paper does not describe the functional aspects clearly enough.
- D. Boneh et al.[3] make abstraction of the marking algorithm and consider cryptographic issues. The authors discuss the main problem of the *fingerprinting* technique.
- B. Pfitzmann et al.[14] describe a global scheme to trace people who abuse broadcast encryption schemes and introduce the interest of fingerprinting. Nevertheless, this paper remains theoretical.

  Finally, there exist now quite a few companies involved in the area of watermarking, such as DIGIMARC, Signum Technologies, R3S, Mediasec Technologies or CRL. Some bigger companies are also currently developing watermarking techniques, such as IBM, AT&T, SONY, NTT, Matsushita, NEC, Philips. It is quite difficult to collect information about their technologies.

3. AQUARELLE FRAMEWORK

3.1. Aquarelle aims and technical objectives

Documentation - in a broad sense - is becoming one of the major productions of museums and cultural organizations. To organize exhibitions and produce information products cultural organizations, museums, libraries, photo-agencies, research laboratories or publishers, have to share information. The aim of Aquarelle is to present a global system for accessing this information.

The main technical objectives of the Aquarelle project are the following: to develop a unified resource discovery system for the cultural heritage information available in archive and folder databases; to provide facilities supporting information access through hypertext navigation as well as information retrieval by querying.

The archive server databases contain the images that we want to protect. We consider that organizations running archive servers either own the images they contain, or distribute images belonging to another entity. In both cases, we consider that the organization running an archive server is willing to protect the images which are on the archive.

The Aquarelle architecture is quite intricated and sophisticated. For our purpose, we only present the following simplification. The front-end user uses a standard browser or the Aquarelle advanced browser. He connects to the "User Client Server WEB", which provides front pages and various cgi-bin. The "access server" is the key entrance to the Aquarelle system. In that place, users are registered, and data is transmitted through that node. Through the Z39.50 protocol, data may come directly from the archive servers, or from the "folder server", where folders contain meta-data, and may be published by publishers or cultural organizations.

3.2. Users requirements and system constraints

Cultural partners were concerned with the possibility that their images could be re-used in an unauthorized way. Their main concern was to able to prove their ownership of images. Such a possibility is a detering threat to potential cheaters.

Because of such an objective, they wanted the watermarking system to present a high level of efficiency for protection. This means that the embedded mark must be very robust. Basic cryptographic commandments imply that the algorithm must be parameterized with some key. So we decided that a unique key would be used for each image.

In the Aquarelle architecture, users connect to the system through an *Aquarelle Access Server*. Once logged they are able to formulate their query. This query is broadcasted to folders and archive servers, and a result set is presented to the user. Connections between users and Access Servers are performed through the HTTP protocol. Connection between Access Servers and core data servers are performed using the Z39.50 protocol. The login and password are managed by the Access Server.

For the Z39.50 connection between the Access Server and the core data Servers, a login and user password is provided. This enables to authenticate the Access Servers with respect to the core data servers. *There is no user authentication at this level.*

Since users are unknown at the archive server level, there is no possibility for fingerprinting images here. Fingerprinting can not beat the access server level, since information cannot be cross compared between users queries and delivered information. Furthermore access servers are seen as being too busy at logging users, managing connections, formulating queries, collecting and assembling results sets. They are not able to perform an expensive on-line operation as fingerprinting. It is also easier to implement off-line watermarking at the archive server using an easy to manage software piece.

4. PROPERTIES OF THE ALGORITHM

The algorithm which we use in the sequel and for our implementation has been designed at UCL, by Jean-François Delaigle and Benoit Macq.

4.1. Rough description

The watermarking technique used here is based on Human Visual System Model that guarantees that the watermarked picture has the same quality as the original. The watermark is a correlation pattern, which has strong correlation properties.

4.1.1. What is added to the image ?

The technique is additive. From the original, an image of the same size is generated. This image, the watermark, contains data that allow to identify copyright ownership. Basically, the watermark is composed of several replication of the same pattern. This redundancy is necessary for robustness purposes. Each part of the watermark is a modulation of a basic pattern. These parts are added together before being processed in order to be invisible. This process will be described in section 4.1.2.

The basic pattern is composed of black and white rectangles of pixels. Each block stands for one bit. Those bits form MLS sequences. MLS sequences are binary sequences having very good correlation properties, since MLS sequences are nearly orthogonal to their shifted versions.[11] This feature is taken into account during the retrieval process.

An additional security feature was added, the sequences bits are pseudo-randomly mixed before being mapped into the rectangles, with the use of a secret key.

4.1.2. How is it added?

Each part of the watermark is generated independently. It is a modulation of a basic pattern at a secret frequency and a secret orientation, determined by a secret key and a pseudo-random generator. For this frequency and this orientation, a perceptual mask is computed. It serves to adjust the level of the modulated pattern to have it invisible when added to the image. This procedure is repeated several times for different frequencies and orientations. The resulting watermark contains classically 16 repetitions of the same pattern.

4.1.3. How is it decoded?

The retrieval procedure is simple. Each part of the watermark is extracted from the watermarked image, by demodulation and filtering, before being added together. The result is an image very correlated with the basic pattern if the watermark was present. Autocorrelations are compared to cross-correlations (correlations with shifted MLS) to determine whether an image has been watermarked.

4.2. Robustness

The robustness is provided by the use of MLS and the perceptual mask that allows to embed at a higher level in high activity regions regions of the image.

compression When the image is compressed at JPEG 10% the watermark can still be recovered, though the quality of the compressed image is very bad.

filtering The watermark is still recovered after low-pass filtering (e.g. blurring 7x7).

printing The watermark is still recovered after half-tone printing. After redigitizing by scanning, we still recover the watermark.

cropping and scanning In this case we need to know the size of the original to retrieve the watermark.

4.3. Invisibility

The quality of the watermarked image is the same as the original image, thanks to the use of perceptual masking.

4.4. Functional aspects

The algorithm as described in the previous subsections, is well suitable for our purpose.

- It is secured by a secret key.
- It is optimized for high quality still pictures.
- The decoding procedure is a Yes or No decision that determines whether the image is watermarked or not.
- The watermark resists perfectly to classical image processing in image editing and distribution.

5. FUNCTIONAL MODELS

5.1. The Trusted Third Party

Following basic cryptography, any watermarking algorithm must be public, but parameterized by some key. In such a way the algorithm can be made public, and all the secrecy resides in the key.

Known embedding algorithms are such that the knowledge of the embedding key K is needed to verify the watermark. We have two ways of using such a property:

1. The owner reveals the key K to a verifier. The verifier runs the decoding algorithm to check that the image has been marked with the key K.
2. The owner does not reveal the key K, and runs the algorithm for himself.

In the first case, the incrusted image is not reusable, since the key K has been shown, and anyone knowing K is able to remove the mark.

In the second case, the owner may be a lier, since from an outside point of view, it seems only that the owner is running a black box which outputs YES. He can not be trusted.

We solve these issues by introducing a Trusted Third Party, the TTP, which plays the following role:

- The TTP knows the secret key K.
- The TTP will never reveal the key K.
- The TTP runs the decoding algorithm, outputs the answer and never lies.

The TTP is also trusted for the following features: it is highly secure, from many points of view (see Section 6.3). The secrets key K can not be violated, and there can be no impersonnification of the TTP.

The TTP introduced here is not a registration authority of copyright-ownership. The TTP will trust the copyright-owners who wish to use its services, and will not check whether the image belongs or does not belong to the copyright-owner using its services.

5.2. Entities

In Aquarelle we will consider the following entities:

TTP: the Trusted Third Party

CO: the owner of the copyright of IM. From the Aquarellepoint of view, we see it as an archive-server manager

CO-ID: a string which is the unique image identifier of CO

IM: the original image

IM-ID: a string which is the unique image identifier of IM

D: the date

IM:* the watermarked image

*IM**:* the watermarked image, eventually modified by some hacker

K-IM: the secret used to perform the embedding for that particular image

User: a sample user of the Aquarelle system

5.3. A first functional model

We present a first functional model for clarity purpose. It is NOT the one which is implemented, but it is useful to understand the next one and its advantages.

The protocol for watermarking using the above algorithm runs in 3 phases:

1. The Copyright-Owner sends IM, IM-ID and CO-ID to the TTP
2. The TTP generates a random key K-IM, watermarks the image with K-IM, and securely keeps (IM-ID,CO-ID,D,K-IM) in a table.
3. The TTP sends the watermarked image IM* back to the Copyright-Owner, along with CO-ID, IM-ID.

The Copyright-Owner may now deliver the watermarked image IM* through the Aquarelle system. The verification phase is as follows:

1. A user submits an image IM**, IM-ID and CO-ID.
2. The TTP replies YES or NO.

The date field in the database of secret keys is introduced to prevent the following scenario. An image-owner CO1 wants to cheat: he picks an image that has been marked by CO at date D, and submits it to the TTP with the identifiers CO1 and IM-ID1 for watermarking. Both CO and CO1 are able to have their watermark checked by the TTP. But since CO1 submitted the image after CO, then the date field D1 related to CO1, IM-ID1 is bigger than the date D from the original query, the fraud can be detected.

But the above watermarking protocol has the two following disadvantages. First the image must be transmitted over a secure line for the first phase, since an eavesdropper may steal the unmarked image, which has no protection at that time. Secure line may mean encryption, which is a difficult issue because of various regulations on that topic in European countries (notably France). The second disadvantage is that there are two exchanges of images between the CO and the TTP, which makes a large amount of data to be transmitted.

The improved watermarking protocol presented below solves these two problems.

5.4. Using the Diffie-Hellman protocol

The improved model uses the Diffie-Hellman key-exchange protocol:[6] it enables two persons to share a common secret, without any secure communication; it gets its security from the difficulty of calculating discrete logarithms in a finite field.

Diffie-Hellman key exchange The Diffie-Hellmann protocol enables two persons Alice and Bob to share a common secret, without any secure communication.

Two integers are publicly known: a prime number p and $g < p$. The computations are done *modulo* p. It is very difficult (impossible) to find a from the data of g^a (this is known as the *discrete logarithm problem*). Since the discrete logarithm problem is as intractable as the factoring problem, and one can say that, from a practical point a view, the Diffie-Hellmann protocol is as secure as RSA.

The protocol is run as follows (see figure 1).

1. Alice randomly generates x_A computes $K_A = g^{x_A}$ and transmits K_A to Bob; Bob randomly generates x_B, computes $K_B = g^{x_B}$ and transmits K_B to Alice.
2. Then Alice computes $K_B^{x_A} = (g^{x_B})^{x_A} = g^{x_A x_B}$, and Bob computes $K_A^{x_B} = (g^{x_A})^{x_B} = g^{x_A x_B}$.
3. Alice and Bob now share a common integer $C = g^{x_A x_B}$, which is unknown to anyone else.

Note that for a prime number p of length 1024, the exchanged data have a length up to 128 bytes, which is very short, for a very secure scheme.

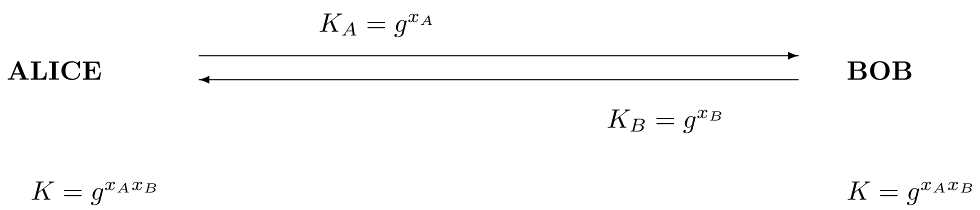

Figure 1. The Diffie Hellmann protocol

Watermarking using Diffie-Hellmann The protocol for watermarking runs in 3 phases (see figure 2).

1. The Copyright-Owner and the TTP share a common secret key K-IM using the Diffie-Hellman protocol (each of them sends to the other his Diffie-Hellman half public key, say K_A for the CO and K_B for the TTP).
2. The TTP securely keeps (IM-ID,CO-ID,D,K-IM) secret.
3. The CO marks the image with the key K-IM.

This protocol is an improvement of the previous one since no images are exchanged between the CO and the TTP, so there is no need for a secure communication. Second the data exchanged between the CO and the TTP is very small, a few thousands bits, say.

We name this protocol the DHWM protocol, standing for "Diffie-Hellman protocol for Water-Marking".

6. SECURITY OF THE SCHEME

6.1. Scenarii

6.1.1. Normal scenario

After running the CO-request-and-WM protocol and posting the image IM* through Aquarelle, the CO discovers that the image IM* is illegally used (on a WEB server, on printed published material etc). He submits CO-ID, IM-ID and IM* to the TTP. **The TTP replies YES.**

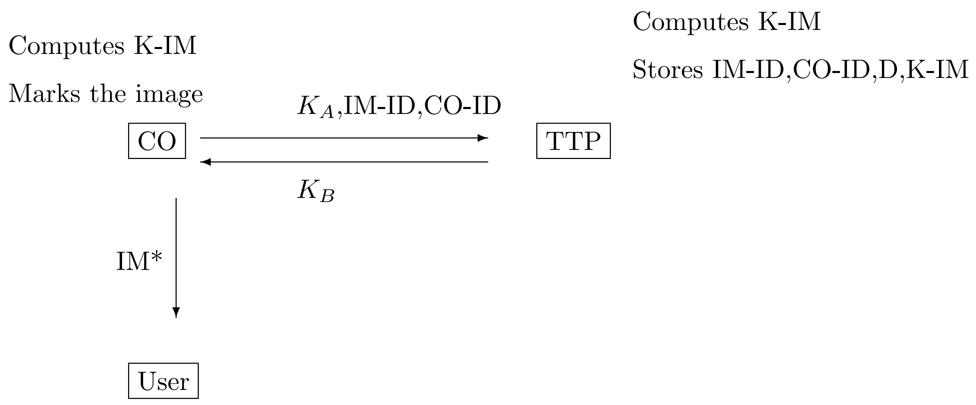

Figure 2. A scheme for watermarking images with the Diffie-Hellman protocol

6.1.2. Hacked watermarked image submitted by CO

Again, after running the CO-request-and-WM protocol and posting the image IM* through Aquarelle, the CO discovers that the image IM* is illegally used. In fact the image has been hacked (by JPEG compression, or by adding noise to the image), and is in fact IM**.

He submits CO-ID, IM-ID and IM** to the TTP. **The TTP should reply YES.** However since there may be a zone of uncertainty for the decoding algorithm, the TTP **must not reply NO**.

6.1.3. The image does not belong to CO

The CO discovers an image IM' and thinks that the image belongs to him, that is, he has run CO-request-and-WM for an image IM, with the fields CO-ID and IM-ID, and IM' is very similar to IM. This could be the case, for example, of a photograph of the Eiffel Tower shot by another photograph, in nearly the same conditions.

Believing that IM' belongs to him, he submits CO-ID, IM-ID, and IM'. **The TTP replies NO** (the image IM' has not been watermarked with the key related to CO-ID and IM-ID).

6.1.4. Image picked by CO'

CO has run the CO-request-and-WM protocol with the fields CO-ID and IM-ID, and has posted the image IM* through Aquarelle. A fraudulous CO' picks the image IM*, does not mark it and stores it in its own database, with the fields CO'-ID and IM-ID'. On dispute, CO is able to have the YES answer from the TTP, and CO' gets the answer "CO'-ID, IM-ID are not in my database" from the TTP, which in that case, does not need to run the verification algorithm. A simple check in the database is enough.

6.1.5. Image furthermore watermarked by CO'

CO has run the CO-request-and-WM protocol with the fields CO-ID and IM-ID, and has posted the image IM* through Aquarelle. A fraudulous CO' picks the image IM*, and runs the CO-request-and-WM protocol with the fields CO'-ID and IM-ID', thus watermarking the image once more. Both CO and CO' are able to get the YES answer from the TTP, but with the date field, the TTP is able to check that the image IM has first been watermarked by CO, thus defeating CO'.

6.1.6. Image hacked and furthermore watermarked by CO'

CO has run the CO-request-and-WM protocol with the fields CO-ID and IM-ID, and has posted the image IM* through Aquarelle. A fraudulous CO' (recognized by the TTP), picks the image IM*, hacks the image into IM** (as in the scenario 6.1.2) and runs the CO-request-and-WM protocol with the fields CO'-ID and IM-ID', thus watermarking the image IM** one more time. CO' post the image IM*** (watermarked, hacked, watermarked again). CO' can easily get the answer YES from the TTP (as in the 6.1.1 scenario). If the mark is robust enough, CO is also able to get the YES answer from the TTP, and the date field enables the TTP do defeat CO'. As in scenario 6.1.2, the TTP **must not reply NO** to CO.

6.2. The fuzzy answer

When dealing with images, there is always some level of uncertainty. The above scenarios show that in some cases, the answer must be a definite YES, a definite NO, or must not be NO. We believe that this must be taken into account for the answer given by the TTP and for the design of the watermarking algorithm.

There will always be a zone where the YES will be a strong YES, and NO will be a strong NO, for most images. Next there is a zone with a confident yes and a confident no. Next, there will be a zone of perhaps yes and perhaps no. We think that the TTP must not venture into giving strong answers when he is not able to do so (this highly depends on the watermarking algorithm).

We propose to introduce a fuzzy answer, which is simply "I do not know". This is important when two COs are conflicting, and the fuzzy answer must induce another resolution to the problem, without using the watermarking technology. For example, both COs may go to court and have independant image experts analyze their images to get some clues.

For a very performant watermarking algorithm, the "fuzzy answer zone" should be as small as possible. But since an image is watermarked for a long life-time, and since future technologies may provide new compression algorithms for images and new theories for the analysis of images, which may defeat the watermarking algorithm, the fuzzy answer is a good escape door.

6.3. Security considerations

There remains three main issues to address in the DWHM protocol.

Random numbers Running the Diffie-Hellman protocol, we need random numbers for generating the Diffie-Hellman half public keys. We chose here to use a self-shrinking generator.[13] Known attacks against this kind of pseudo-random generator only apply when the opponent is able to look at a very long string of bits. Here, we only need small pseudo-random strings for our protocol, and we are then protected against the attacks on our generator. Moreover, it is very fast, and this is an important point since the TTP could perform the random numbers generator very often. The weak point is that the state of the machine must be stored in a file, and attacks on this file may be considered. So this file (`DH_seed` in our implementation) must be protected.

Security of the TTP database It is more obvious that the secrets maintained by the TTP must not be discovered by anyone. A cryptographic solution may consist in encrypting the IM-ID field in the database with a key only known by the TTP, but since the TTP acts automatically, this key must be stored somewhere. So the problem of protecting the file where the key is stored still remains.

We believe that this problem is more related to computer security than to cryptology. In our implementation, the file is simply protected by usual Unix rights, and only the HTTP server is able to read this file.

We leave the problem to computer security specialists, and suggest to use specialized software for this issue. We also suggest to limit the Internet Protocols that are used by the TTP.

Authentication While the Diffie-Hellman protocol is designed to be protected against an eavesdropper (i.e. a passive attack), it does not offer protection against active attacks. We mainly think of authentication: it is a main issue that the CO and the TTP can be ABSOLUTELY assured of each other identity when they run the DHWM protocol to share a common secret key.

Since the protocol is built onto the HTTP/1.1 protocol, any security tool or software for authentication for the World Wide Web is convenient here. We think that the COs must be registered by the TTP, and the TTP must not be subject to an impersonnification attack.

This issue is not covered by the DHWM scheme. Many authentication schemes are proposed for the WEB and for the HTTP protocol. HTTP/1.1 provides a better authentication tool than HTTP/1.0 (login,password). SSL protocol also enables authentication, with heavier tools.

Nevertheless, the Diffie-Hellman protocol can integrate authentication, using a three-round protocol instead of a two-round protocol.[7] This protocol combines Diffie-Hellman key exchange and authentication, and is the basis of the Photuris protocol for IP security.

7. CONCLUSION

In a system like Aquarelle, where images are distributed to registered users, we protect the images with a watermarking solution. The chosen algorithm from the catholic University of Louvain has very good properties with respect to robustness, invisibility, resistance to JPEG compression. It offers a low functionality, since it actually embeds a single bit of information in an image. Using the DHWM scheme, this small amount of information is turned into a copyright protection system, using a Trusted Third Party.

A simple implementation using an HTTP server has been made. This enables to use any emerging security tool for the WEB to improve the security of the model.

REFERENCES

1. J. Brassil ans S. Low, N. Maxemchuck, and L. O'Gorman. Electronic marking and identification techniques to discourage document copying. In *IEEE INFOCOM'94 - Networking for global communications*, 1994.
2. F.M. Boland, J.J.K. O' Ruanaidh, and C. Dautzenberg. Watermarking digital images for copyright protection. *Image Processing and its Applications*, pages 326–330, 1995. July.
3. D. Boneh and J. Shaw. Collusion-secure fingerprinting for digital data. In D. Coppersmith, editor, *CRYPTO 95*, number 963 in Lecture Notes in Computer Science, pages 452–564. Springer, 1995.
4. G. W. Braudaway, K. A. Magerlein, and F. Mintzer. Protecting publicly-available images with a visible image watermark. Technical Report RC 20336 (89918) 1/15/96, IBM Research Division, 1996.
5. I. J. Cox, J. Killian, T. Leighton, and T. Shamoon. Secure spread spectrum watermarking for multimedia. Technical Report 95-10, NEC Research Institute, 1995.
6. W. Diffie and M.E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, 1976.
7. W. Diffie, P. C. van Oorshot, and M. J Wiener. Authentication and authenticated key exchanges. *Designs, Codes and Cryptography*, 2:107–125, 1992.
8. R. C. Dixon. *Spread Spectrum with Commercial Applications*. Wiley, third edition edition, 1994.
9. David Kahn. The history of cryptography. In R. Anderson, editor, *Information Hiding*, number 1174 in Lecture Notes in Computer Science, pages 1–5. Springer, 1996.
10. E. Koch and J. Zhao. Towards robust and hidden image copyright labeling. In *Proc of 1995 IEEE Workshop on Nonlinear Signal and Image Processing*, 1995.
11. R. Lidl and H. Niederreiter. *Finite Fields*. Number 20 in Encyclopedia of Mathematics and its Applications. Addison-Wesley, first edition, 1983.
12. K. Matsui and K. Tanaka. Video-Stenography: How to embed a Signature in a Picture. *IMA Intellectual Property Proceedings*, 1(1):187–205, January 1994.
13. W. Meier and O. Staffelbach. The self shrinking generator. In R.E. Blahut et al, editor, *Communications and Cryptography: Two sides of One Tapestry*, pages 287–295. Kluwer Academic Publishers, 1994.
14. B. Pfitzmann and M. Schunter. Asymmetric fingerprinting. In U. Maurer, editor, *EUROCRYPT 96*, number 1070 in Lecture Notes in Computer Science, pages 84–95. Springer, 1996.
15. R.G. van Schyndel, A.Z. Tirkel, and C.F. Osborne. Towards a robust digital watermark. In S.Z. Li, D.P. Mital, E.K. Teoh, and H.Wan, editors, *ACCV'95, Second Asian Conference on Computer Vision*, number 1035 in Lecture Notes in Computer Science, pages 504–508. Springer, 1995.
16. N. Morimo, W. Bender, D. Gruhl. Techniques for data hiding. In *Proceedings of the SPIE*, February 1995. San Jose CA.
17. J. Zhao and E. Koch. Embedding robust labels into images for copyright protection. In Peter Paul Klaus Brunnstein, editor, *KnowRight 1995*, Schriftenreihe der sterreichischen Computer Gesellschaft, Band 82, pages 242–251. Oldenbourg Verlag, 1995.