ICICS-PCM 2003
15-18 December 2003
Singapore

Dynamic Watermarking of Images

Monika Bansal, Wei-Qi Yan, Mohan S Kankanhalli
Department of Computer Science
School of Computing
National University of Singapore, Singapore 119260

Abstract

With the rapid growth of networked multimedia data systems, copyright protection of proprietary digitized media has gained importance. Inserting a robust and invisible signal that clearly identifies the owner or the recipient is beginning to emerge as the solution. Previous research in the field of watermarking has been successful in inserting a 'static' watermark, which endures ownership rights but is not as robust and tamper-proof as the 'dynamic' watermarks. This paper presents a novel invisible and robust watermarking technique that aims at dynamically inserting the watermark in digital images. Dynamic generation of watermark ensures that whenever the image is viewed, the watermark inserted in the image is different from the previous one providing more security against copyright attacks. This is accomplished by bundling the viewer and the image together, in which the viewer is responsible for embedding the new watermark using the spread spectrum watermarking algorithm every time. We have implemented the proposed scheme and present experimental results.

1. Introduction

This proliferation of digitized media (audio, image and video) is creating a pressing need for copyright enforcement schemes that protect copyright ownership [1]. A large amount of research work has been done in the field of embedding and extracting watermarks that identify the owner and protect the copyrights [2-9]. Although a lot of work is done in the field of watermarking, little or no specific work is done in the area of 'dynamic watermarking', which aims at embedding dynamic information that can be specifically created for the user downloading the image. With the ever-increasing growth in the processing power of image servers and available bandwidth on the Internet for image delivery, together with improved watermarking techniques, dynamic watermarking can be a solution for many types of digital right management scenarios.

In remotely accessed computer systems, a user identifies himself to the system by sending a secret password. There are many ways an intruder could learn the user's secret password and then impersonate him when interacting with the system. To counter this, a scheme was proposed by Lamport [11]. Essentially, the Lamport's algorithm is cited as follows: select a value m, which denotes the number of times a user can log on to the system, select a one-way function F, such that it is easy to compute $F^m(x)$ for some fixed word x, where F^m denotes m successive applications of F on x; with a fixed value of m, the i^{th} password x_i is equal to $F^{m-i}(x)$, where x is the secret password given to the user during registration time. Thus the sequence of m passwords is $F^{m-1}(x)$ **…..**, $F(F(x))$, $F(x)$, x. For a password x', system authenticates the password by applying the same one-way function F to the password and calculating $y'=F(x')$. Thus, the sequence of y_i's needed by the system to store to authenticate the above passwords is $F^m(x)$.... $F(F(F(x)))$, $F(F(x))$, $F(x)$.

Harn [9] proposes another scheme which integrates the concept of public key cryptography [13, 14] and the above dynamic password scheme. Instead of using one-way function, they propose to use a one-way trapdoor, such as digital signature scheme to bind each user's public *ID* and dynamic password together. This paper uses this scheme to generate dynamic watermark strings.

Several algorithms to embed digital watermark have been proposed till now. Caronn [3] suggests adding tags – small geometric patterns – to digitized images at brightness levels that are imperceptible. While the idea of hiding a spatial watermark in an image is fundamentally sound, this scheme is susceptible to attack by filtering and redigitalization. The fainter such watermarks are the more susceptible they are to such attacks. Moreover, this scheme is not applicable to images having uniform brightness at all pixels and may not be robust to common geometric distortions, especially cropping.

Macq and Quisquater [4] provide a description of the procedure to insert a watermark into the least significant bits of a pixel located in the vicinity of image contours. Since, it relies on modifications of the least significant bits, the watermark is easily destroyed. Furthermore, this method is restricted to images, in that it seeks to insert the watermark into image regions that lie on the edge of the contours.

On the basis of various threats to watermarked images, Cox et al [5] argue that a watermark must be placed in perceptually significant components of a signal if it is to be robust to common signal distortions and malicious attack. They suggest a scheme to embed the watermark in frequency domain than in the spatial domain named as spread spectrum coding of the watermark, which is conceived by analogy to spread spectrum communication [15]. This watermarking scheme is known to be robust to several kinds of malicious attacks and thus is used to embed the dynamic password in this paper.

2. Dynamic Watermarking

This section explains the algorithms implemented for generating dynamic watermark string, embedding the watermark and extracting the embedded watermark in detail. We first state the mechanism of dynamic password scheme, then we describe the watermarking scheme.

2.1 Dynamic Password Scheme

The dynamic password scheme explained in this paper used the scheme defined by Harn [10], which integrates the concept of public key idea [14] and the dynamic password scheme [11]. Step-wise implementation of the dynamic password generation scheme proposed by Harn [11] is pictorially described in figure 1.

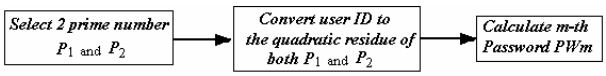

Figure. 1 Three-step algorithm to generate dynamic password

Algorithm: (Generating dynamic password)

Step 1: The system selects two large distinct prime numbers p_1 and p_2 of the form $4m+3$ (where m is a natural number), and compute $n= p_1 p_2$.

Step 2: During registration, user i will obtain a unique ID_i from the system.

Step 3: The system computes the m-*th* password PW_m according to the equation

$$PM_m^{2\wedge m} = ID_i^{'} \bmod n \quad (1)$$

It is necessary to make the ID_i a quadratic residue of both p_1 and p_2. The relationship between a given ID_i and a modified ID_i (written as ID_i') is denoted as (2).

$$ID_i^{'} = ID_i \cdot k \quad (2)$$

In (2), ID_i' belongs to the intersection of the sets $QR^* \ p_1$ (quadratic residue set of prime p_1) and $QR^* \ p_2$ (quadratic residue set of prime p_2) and k can be either 1, 2, -2 or -1. Thus, any ID_i can be converted into the required ID_i' by the above relationship. Some value of k belonging to {1,2,-2,-1}, will convert any user ID to the quadratic residue of both p_1 and p_2. Associated with each ID there is another entry consisting of two bits: j and h. During real-time login upon receiving user i's ID, these two bits will tell the system which modifier, 1, 2, -2 or -1 should be used to modify ID_i to ID_i'. Corresponding to all users there is a login time stored, which tells the system of how many times user has logged on to the system. This login time is updated every time user logs on to the system. User's login time is set to zero at the beginning, i.e. first password is entered.

For the first password is solved with m = 1.

$$PM_1 = ID_i^{'} \bmod n \quad (3)$$

Since ID_i' is a quadratic residue of both p_1 and p_2, the above equation is solvable generating the first password PW_1. The distinct PW_1 belongs to the intersection of the sets $QR^* \ p_1$ (quadratic residue set of prime p_1) and $QR^* \ p_2$ (quadratic residue set of prime p_2). Similarly, PW_2 , second login password can be generated by solving the (3) with m=2.

$$PM_2^{2\wedge 2} = ID_i^{'} \bmod n \quad (4)$$

Comparing (3) and (4) we get, $(PW_2)^{2\wedge 2} = ID_i' \bmod n = (PW_1)^2 \bmod n$, implying by continuing the above process for m times, a sequence of passwords $PW_1, PW_2 \ldots PW_m$ can be obtained.

$$PM_2 = PM_1 \bmod n \quad (5)$$

While logging in for the r-*th* time, user i submits his public ID_i, issued to him during registration and computes the dynamic password PW_r, as

$$PW_r = (PW_m)^{2\wedge(m-r)} \bmod n, \text{ where } 1 \le m \le r \quad (6)$$

The system first checks the validity of user i's ID. If it is a valid user ID, the system will do the following:

1) Examine bits j and h corresponding to ID_i to modify it to ID_i'.
2) Retrieve the login time, s (which is equal to r-1), from the table corresponding to user i and compute $(PW_r)^{2\wedge(s+1)} \bmod n$. If this computed result matches with the public ID_i', login succeeds. For a valid password this will hold.
3) Update the login time s of the user i from the table by increasing one.

Following features of the scheme explained above makes it robust, efficient and hence useful for copyright protection.

1) The security of this scheme depends on the difficulty of factoring n into two large prime numbers p_1 and p_2. Rivest et al [16] suggest that the scheme can be robust on using 200 digit long prime numbers. In addition, both p_1-1 and p_2-1 should have large prime factors and gcd (p_1-1, p_2-1) should be small.
2) This scheme uses the digital signature scheme to bind each user's password and ID. Thus, there is no need to store the passwords or encrypted passwords on the authentication server.
3) The time required for each user to generate r-th login password is (m-r) modular multiplications. The time required by the system to verify the submitted password for r-th login is r modular multiplications. Thus, the overall time required for login operation is 'm' modular multiplications compared to other secured one way functions (as RSA), which generally involve a number of modular exponentials. Hence, this method is much more time efficient.
4) There is no secret information to be used by the system during the authentication process.
5) Since the user's password is changed dynamically, it is almost impossible for an eavesdropper to tamper with the communication and to impersonate the user.
6) Revoking the user's access privilege can be easily achieved by removing user's ID from the system.

However, each user's password has to be very long as a short password is very vulnerable to various kinds of attacks, which can be difficult to be memorized. This problem can be solved by storing the password in a portable diskette or a magnetic card or by generating the larger password pseudo-randomly from a smaller seed.

2.2 Watermarking Algorithm

Though any of the robust watermarking algorithms can be used, the dynamic watermarking scheme proposed in this paper uses spread spectrum watermarking algorithm proposed by Cox et al [5] which embeds the watermark in the frequency domain and has been proved to be robust to all the common threats to watermark.

Algorithm (Step-wise watermarking procedure)**:**

Step 1: Reading the original image;
Step 2: Taking the frequency (DCT) transform;
Step 3: Inserting the watermark;
Step 4: Taking the inverse frequency transform;
Step 5: Comparing watermarked images;
Step6: Evaluating the similarity of two watermarked images.

From the image *I*, for all the blocks perceptually significant sequence of values $V = (v_1, v_2,\ldots\ldots v_n)^T$ is extracted into which the watermark $X = (x_1, x_2,\ldots\ldots x_n)^T$ is inserted in the frequency domain to obtain an adjusted sequence of values $V' = (v'_1, v'_2,\ldots\ldots v'_n)^T$ which is inserted back into the image in place of *V* to obtain a watermarked image *I'*. While inserting *X* into *V* to obtain *V'*, a scaling factor α is specified which determines the extent to which *X* alters *V*. As stated by Cox et al, three approaches can be used to compute *V'*. In the implemented scheme the following equation was used.

$$v_i' = v_i(1 + \alpha x_i) \quad (7)$$

2.3 Implementation of the Executable Package

When owner *A* intends to render an image for use on Internet, instead of the sharing the watermarked image, he renders the corresponding executable package. First, a user who intends to view the image has to register himself with the owner *A*, when he is issued a user *ID*. Based on this used *ID* the dynamic password scheme generates the dynamic password, changes it into the corresponding binary string and generating the watermark string by changing the 0's with –1's. This watermark string is spread by the spread factor called chip rate (*cr*) based on the number of pixels of the original image that are to be watermarked as shown in (7). The expanded string is embedded in the original image using the spread spectrum watermarking algorithm.

$$\text{Chip Rate } (cr) = \frac{\text{Number of pixels to be Watermarked}}{\text{Length of the Watermark } (n)} \quad (8)$$

Thus, the executable package is created by bundling the dynamic password scheme and spread spectrum watermarking algorithm with an image viewer tool. There are some data files having the image and user information, which are encrypted and stored in the executable package. When the executable file is run, the encrypted data files are decrypted, the dynamic password scheme calculates the next watermark string which is embedded using the spread-spectrum watermarking technique. The viewer bundled with the executable package displays the watermarked image in a pop-up window. When user closes the pop-up window the executable package increments the number of times the image is viewed and updates the relevant data file. It encrypts the data files and stores them in the executable package. When this file is run again, a new watermark string is computed and the same procedure is repeated.

To extract the watermark, the input original image and the watermarked image are both changed to frequency domain using the Discrete Cosine Transform (DCT) to get v_i and v_i'. Since the watermark is spread all over the image the integration can be done only after selecting the different pixel locations where the watermark $X = (x_1, x_2,\ldots\ldots x_{n*cr})^T$ is embedded. On selecting such pixel locations the corresponding v_i is subtracted from it getting $v_i\alpha\ x_i$ by equation (7). The objective of the extraction algorithm is to detect the value of x_i from this value without knowing the value of alpha α. As the watermark string is a sequence of 1's and -1's and each bit is spread chip rate *cr* times, therefore, summation of *cr* number of pixels will either return a positive or a negative value. Thus, on integrating this value over the chip rate, the sign of the sum indicates the embedded watermark bit.

2.4 Obfuscating Class Files

As mentioned earlier, for the robustness of the executable package, the class files are obfuscated using the software *Smokescreen* [17]. Figure 2 shows a snapshot of the software interface. The classes to be obfuscated are placed in a source directory and the mangled classes are written in the destination directory. Furthermore, it has the option to select the level of obfuscation desired, ability to change the control flow and adding machine language statements that makes the de-compilation of the class files almost impossible.

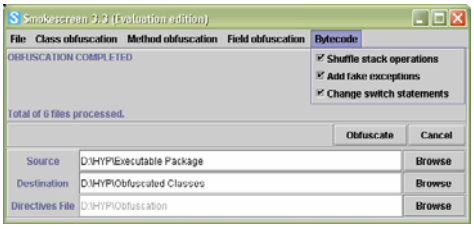

Figure.2 Snapshot of Java obfuscator smokescreen

3. Experiments

This section describes the various implementation issues and discusses the approach taken on the basis of experimental observations. This section also discusses various issues related to implementation and lists experimental results achieved. Issues related to dynamic generation of the watermark string and spread-spectrum watermarking algorithms are discussed by taking one sample input. Once an executable package is created corresponding to the sample image, a sample watermarked image is shown.

3.1 Generating Dynamic Watermark Strings

Section 3.1 and 3.2 explain the structure of the watermark string and the algorithm used to generate it. Using the algorithm with sample input of p_1= 139, p_2 = 199, user ID = 20, the first 10 passwords can be calculated which are listed in Table 1.

Table 1: List of 10 dynamically generated passwords

Number of Times user has viewed the Image	Dynamic Password	Dynamically generated Watermark String
0	7808	1 1 1 1 -1 1 -1 -1 -1 -1 -1 -1 -1 -1
1	3315	1 1 -1 -1 1 1 1 1 -1 -1 1 1
2	176	1 -1 1 1 -1 -1 -1 -1
3	4512	1 -1 -1 -1 1 1 -1 1 -1 -1 -1 -1 -1
4	7906	1 1 1 1 -1 1 1 1 -1 -1 -1 1 -1
5	7208	1 1 1 -1 -1 -1 -1 1 -1 1 -1 -1 -1
6	8086	1 1 1 1 1 1 -1 -1 1 -1 1 1 1 -1
7	2342	1 -1 -1 1 -1 -1 1 -1 -1 1 1 -1
8	1486	1 -1 1 1 1 -1 -1 1 1 1 -1
9	6518	1 1 -1 -1 1 -1 1 1 1 -1 1 1 -1

In case of copyright violation, the embedded watermark string is extracted from the image and the corresponding user ID from which was generated using the Dynamic Password Scheme can be obtained. It is to be noted that for different user ID's the passwords generated are unique as the extraction process returns a unique value always.

3.2 Detecting Perceptually-significant Portions of an Image

According to the algorithm proposed by Cox [5], the watermark string should be placed in perceptually significant portion of the image. The image is divided into 8 by 8 blocks. Once the image is transformed to frequency domain from spatial domain, there are mainly four portions in a block that can be watermarked as demarcated in Figure 3. Region 1 and 2 are low frequency regions, Region 3 forms middle frequency region and Region 4 consists of the highest frequency bits of an image spectrum. These four regions, the watermark is placed in the region which forms a significant portion of the image spectrum and with the insertion of the watermark string in that region, the Peak Signal to Noise Ratio (PSNR) does not fall. Keeping this in mind, watermark is embedded in all the four regions using (7) with different values of variable alpha α and experimental results are collected. On the basis of these results, Region 2 is selected for efficiently embedding the watermark bits in the image. These experiments are carried out region wise and the results are given in the following sub-sections.

The 8x8 block is divided into 4 regions and watermark string is embedded in them and the quality of the watermarked image is compared using PSNR values which are depicted in the following graph.

The above Figure 4 shows that the red region which forms the region of high frequency give the best PSNR value, but it is observed that such pixels do not form the perceptually significant region. Watermark embedded in this region is not robust to attacks from compression. Hence, the second best region of low frequency which forms the perceptually significant region is chosen and the selected pixels are watermarked.

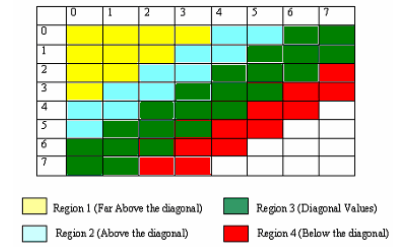

Figure.3 Diagrammatic representation of an 8 by 8 block. The figure highlights the regions where the watermark can be embedded with different shades.

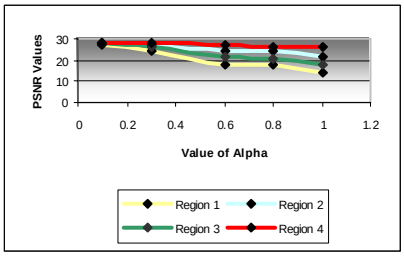

Figure. 4 Graphical representation of the data

3.3 Watermarking an Image Using the Executable Package

The executable package has been tested rigorously by embedding and extracting the watermark string with a wide range of input values. A registered user gets access to the executable package, which when run displays the watermarked image. Figure 5 shows both the original and the watermarked image. Region 2 is watermarked with a value of alpha, α equal to 0.3. 26-bit long watermark string is taken and is spread by a chip rate. PSNR value computed for the following figure is 27.17 db.

Figure.5 Embedded a dynamic watermarking the image

Figure 5 is a sample image watermarked using the executable package and undergone JPEG compression. The 26-bit long watermark string embedded is X = (-1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1) ^{T}. The extraction algorithm is used to detect the embedded watermark string and is compared with the original one by using (7). The extracted watermark string from the above sample image is X^* = (-1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 1 -1 1 -1 1 -1 1 1 1) ^{T}. The difference between the

embedded and the extracted watermark vectors is computed to be 0.923.

4. Conclusion and Future Work

The approach of dynamic watermarking implemented in this paper is to create an executable package corresponding to an input image. This executable package is responsible for dynamically generating the watermark string, embedding it in the original image and displaying the watermarked image to the user. Due to this feature of the executable package, some extra security measures (like encryption of data files and obfuscation of class files) are implemented which aim to make this executable package robust to malicious attacks. The last phase included implementing an extraction algorithm to detect the watermark from a watermarked image. This watermark should be able to prove the ownership rights in case of prosecution of copying of digitalized data.

Work in this research is far from over. There have been some limitations to all the proposed and implemented schemes for protecting the copyrights of digitalized media from malicious users. Once the watermarked image is rendered for use on file-sharing platforms and the Internet, there are many ways in which the digital data can be acquired illegally, manipulated and copied. Total protection of copyrights from all such means has always remained a challenge. One of the major threats to the dynamic watermarking scheme is when the user is able to save the watermarked image. When the executable package displays the watermarked image, malicious users can save it by taking a snap-shot of the screen or capturing the photo in a digital camera. Once the image is copied, there is no way to embed the dynamic watermark. Limitations and threats to obfuscation of class files (used in the creation of the executable package) are added to the limitation of the package also.

Further research work in the field of dynamic watermarking can involve developing of techniques to provide a tamper-proof way of embedding the watermark dynamically every time image is viewed by a user which are robust to the threats mentioned above. Moreover, these techniques should not increase the processing time of the images as dynamic watermarks are mostly embedded on-the-fly during download from the Internet. Future work in this field would require embedding a dynamic watermark in digitized data of audio and video also.

References

[1] C. P. Pfleeger, Security in Computing (Second Edition), New York: Prentice Hall, Inc. A Simon and Schuster Company, 1997.

[2] C.I. Podilchuk and W. Zeng, “Perceptual watermarking of still images”, Proceedings of First IEEE Signal Processing Society Workshop on Multimedia signal Processing, June 1997

[3] G. Caronni, “Assuring ownership rights for digital images”, In Proceedings of Reliable IT Systems, VIS’95, Vieweg Publishing Company, 1995.

[4] B.M. Macq and J.J. Quisquater, “Cryptography for digital TV broadcasting”. Proceedings of the IEEE, 83 (6): 944-957,1995

[5] I. J. Cox, J. Kilian, T. Leighton and T. Shamoon, “Secure spread spectrum watermarking for multimedia”, IEEE Journal on Selected Areas of Communication, 16(4), pp.587-593, 1998.

[6] I. Pitas, “A brief report on watermarking methods”, Artificial Intelligence Information Analysis Laboratory, Department of Informatics, Aristotle University of Thessaloniki, 2001. https://www.cerias.purdue.edu/infosec/bibtex_archive/archive/2002-30.pdf

[7] J. Fridrich, “Combining low-frequency and spread-spectrum watermarking”, Centre for Intelligent system, Mission Research Corporation, 2001. http://citeseer.nj.nec.com/ fridrich98combining.html

[8] M. Kankanhalli, S.P. Mohanty and K.R Ramakrishnan, “A dual watermarking technique for images”, Proceedings of the 7th ACM International Multimedia Conference, Part II, pp: 49-51, Orlando, Florida, November 1999.

[9] R. Chandramouli, Benjamin M. Graubard, Colin r. Richmond, “A multiple description framework for oblivious watermarking”, Proceedings of Security and Watermarking of Multimedia Contents III, PIE, Vol. 4314, 2001.

[10] Lein Harn, “A public-key based dynamic password scheme”. Symposium on Applied Computing, pp: 430-435, 1991.

[11] L. Lamport, “Password authentication with insecure communication”, Communications of the ACM, 24(11), pp. 770-774, November, 1981.

[12] H.J. Beker, G.M Cole, “Message authentication and dynamic passwords”, Advances in Cryptology, EUROCRYPT, 1987.

[13] R.L Rivest, A.Shamir and L.Adelman, “A method for obtaining digital signatures and public-key cryptosystem”, Communications of the ACM, 21(2), pp. 120-126, February 1978.

[14] T. ElGamal, “A public-key cryptosystem and a signature scheme based on discrete logarithms”, IEEE Transaction Information Theory, Vol. IT-31, pp. 469-472, 1985.

[14] R.C Merkle and M.E. Hellman, “Hiding information and signatures in trapdoor knapsack”, IEEE Transaction Information Theory, IT-24(11), pp. 770-774, November, 1981.

[15] R.L. Pickholtz, D.L. Schilling and L.B. Millstein. “Theory of spread spectrum communiations - a tutorial”, IEEE transaction on communications, pp. 855-884, 1982.

[16] R.L Rivest, A.Shamir and L.Adelman, “A method for obtaining digital signatures and public-key cryptosystem”, Communications of the ACM, 21(2), pp.120-126, February 1978.

[17] http://www.leesw.com/