Data Protection Using Watermarking in E-Business

Alexander P. Pons and Hassan Aljifri, University of Miami, USA

ABSTRACT

In the past decade, the business community has embraced the capabilities of the Internet for a multitude of services that involve access to data and information. Of particular concern to these businesses have been the protection and authentication of digital data as it is distributed electronically. This paper proposes a novel approach that combines the reactive rule-based scheme of an active database management system (ADBMS) with the technology of digital watermarking to automatically protect digital data. The ADMBS technology facilitates the establishment of Event-Condition-Action (ECA) rules that define the actions to be triggered by events under certain conditions. These actions consist of the generation of unique watermarks and the tagging of digital data with unique signatures. Watermarking is a technology that embeds, within the digital data's context, information identifying its owner and/or creator. The integration of these two technologies is a powerful mechanism for protecting digital data in a consistent and formal manner with applications in e-business in establishing and authenticating the ownership of images, audio, video, and other digital materials.

Keywords: active database, watermarking, copyright protection, digital fingerprinting

INTRODUCTION

The Internet has emerged as one of the most profound social, technical, and business phenomena in the history of mankind. It has transformed business (e.g., e-commerce), altered the way individuals communicate (e.g., e-mail), and enabled organizations and individuals access to a wide spectrum and wealth of easily accessible digital data. In e-business, a significant amount of digital data in the form of images, audio, and video continues to be developed and made available to a vast audience. These digital items are referred to as objects. As this trend continues to grow, restrictions on an object's use, authenticity, and ownership are highly desirable, and in some cases, necessary for many companies. Through the use of digital watermarking technology, companies can embed in an object a distinctive signature that uniquely identifies them. The embedded digital watermark can identify an object's owner and/or fingerprint the object and link it to a requestor. Additionally, an object's authenticity is verifiable by utilizing the digital watermark to detect any possible object tampering or alteration. Digital watermarking offers a way for the company to distinctively sign an object, indisputably verifying its ownership and the potential to identify violators, through the embedding of identifiable markings within the object. For example, when a company makes an object available on its Web site, Internet users can download the object to their local machines. These Web clients can use the object in any way they desire including claiming ownership, altering its content, and/or passing the object to others. However, with digital watermarking, the company still would be able to claim ownership, verify the object's content, and determine a violator, since the object contains their identifiable markings.

Numerous areas of e-business have embraced database technology to organize and manage many of these objects. These passive databases function as large object repositories, which render efficient access and management of these objects. Passive databases can be extended using rules and related procedures, which will execute once an object is stored, manipulated, or retrieved, in order to watermark it in a dynamic and unique manner. These active databases respond to object manipulations in ways that enforce established business policies and procedures. The combination of these two technologies, active database and digital watermarking, enables the implementation of an Active Watermarking System (AWS) to protect, track, and authenticate digital data. The proposed AWS (Pons and Aljifri, 2002) automatically watermarks objects that are stored in the database in order to identify the object's owner. When the object is retrieved, it is also watermarked with the requestor's identity to track its release. In addition, the AWS extracts embedded watermarks from an object to authenticate its content and/or to determine the object's owner and/or the object requestor. Organizations and individuals that embrace e-business can greatly benefit from this type of data protection.

The protection of intellectual digital property has gained significant attention in recent years with the 1996 World Intellectual Property Organization (WIPO, 1996a) conference that revised the Berne Convention for the Protection of Literary and Artistic Works to include digital dissemination and use of literary and artistic properties. Provisions of the resulting WIPO Copyright Treaty include several important issues related to future expansion of the use of watermarking techniques. The ideal electronic copyright management system has been described by the writers to include several vital capabilities, including the detection, prevention, and tracking of a number of performed operational functions like opening, printing, copying, or modifying of copyrighted properties (WIPO, 1996b; Burns, 1996; Stefik, 1996, 1997; Smith and Webber, 1995). The AWS supports many of these vital capabilities in a consistent and effective manner through the application of active rules.

The remainder of this paper is organized as follows. The next two sections respectively review the technologies of active database and digital watermarking. Then, we discuss the functionality and objectives of the AWS, and follow up with a section that focuses on AWS implementation issues. Next, the performance of the AWS under various loads is discussed. Finally, we present future enhancements to the AWS and concluding remarks.

ACTIVE DATABASE TECHNOLOGY

Most business applications typically utilize conventional database management systems (DBMS) that are passive and function primarily as data repositories with querying tools to manipulate the stored data. These systems utilize a DBMS despite its inefficiencies and unreliability with regard to the enforcement and consistency of business rules, which reside in external components of the application apart from the database. The placement of business rule processing in external components severely limits their changeability, as all components that enforce the rules are affected, and must be updated individually in order to maintain appliance uniformity. An active DBMS (Widom and Ceri, 1996) provides all of the functionality associated with a passive DBMS and processes business rules in the DBMS by automatically responding to predefined situations or events (inserts, deletes, updates, and queries). When these events occur, conditions (object type, value ranges, etc.) are checked for relevance and if relevant, prompt actions in response to the instigating situation or event. The inclusion of Event-Condition-Action (ECA) rules in a passive database transfers data processing intelligence into the DBMS itself.

The AWS presented in this paper employs active database to enforce copyright protection and traitor tracking for digital media through the establishment execution of certain rules. Consider the AWS rule shown below.

During an insertion into the *Object* table, Rule 1:WM_Image is triggered. The rule determines the type and features of the object. If the object is a JPEG image with features {f_1, f_2, …}, then the system watermarks the image using the appropriate algorithm. The placement of these rules in the DBMS guarantees that the rules and the data are consistent, since a rule is specified once and in one location, instead of residing in each application. This method is ideal for e-business applications, as it solves many rule consistency problems, while potentially increasing performance with the integration of data and rules. Furthermore, two advantages of active databases include: (1) the reusability of rules which reduces the time necessary for the creation and maintenance of rules, and (2) the existence of rule development tools, available in several commercial databases that facilitate rule creation, debugging, and testing to expedite rule implementation.

THE TECHNOLOGY OF WATERMARKING

Digital watermarking is a cutting-edge technology that combines traditional hardcopy watermarking techniques with digital representation. In this section we begin with a survey of watermarking functionality and moves to provide information regarding its application and significance.

Proprietary material often is visually identified with the use of a visible watermark, an insertion or overlaying of a pattern, insignia, or some special identifying mark on or within an object. For example, the fictitious site name www.my-watermark.com might be overlaid on an image created for a Web site banner for marketing purposes, or the United Nations

Rule No: Rule Name	Event	Condition	Action
Rule 1: WM_Image	Inserts into table Object	If object is JPEG image with features {f_1, f_2 …}	Executes image watermarking algorithm

logo might be added to a picture taken at a conference and posted on the Web. The utilization of watermarks in the AWS focuses on watermarks that are not visually identifiable and are generally undetectable to the human eye. These watermarks are secretive, providing a more security-based application of the technology. In addition, distinct to spread spectrum or other steganographic approaches, these watermarking techniques have greater robustness in the sense that the watermark is difficult to extract without altering or degrading the original object.

Watermarking Principles

Since the early 1990s, a variety of watermarking techniques and algorithms have been developed or proposed from a range of communities such as steganography, communications, and source coding. Watermarking systems contain two essential building blocks (Kutter and Petitcolas, 1999): a watermark embedding system and a watermark detection system. Figure 1 shows the general form of a watermarking system. The input to the embedding system consists of a watermark, an object, and a key. The watermark can be in the form of a number, text, or an image. The key enforces security through encryption, preventing unauthorized parties from recovering and manipulating the watermark. The output of the embedding system is the watermarked object.

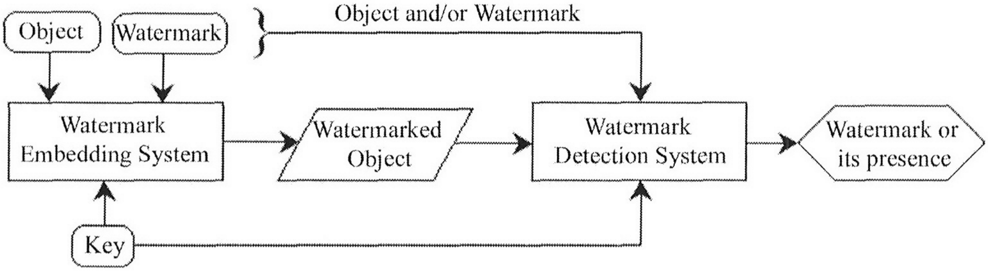
The input to the watermarking detection system contains the watermarked object, the key, and, depending on the watermarking methods, the original watermark or the original object. The output is the detected watermark or an indication of its presence.

Several aspects of an effective and relatively secure watermarking system must be considered: (1) the robustness of the watermark against attacks, (2) the degradation of the data itself in the watermarking process, and (3) the ratio between the host signal and watermark (Katzenbeisser and Petitcolas, 2000). These aspects bring to light perhaps the most important limitation of watermarking — there is a general tradeoff between robustness, perceptibility, and ratio, suggesting that algorithmic design should be highly dependent on the maximization of all three areas, measured independently and against one another.

Watermarking Applications

The requirements with which watermarking systems must always comply are based on the watermarking applications (Voyatzis and Pitas, 1999). It should

Figure 1: Watermarking System


Object
Watermark
Object and/or Watermark
Watermark Embedding System
Watermarked Object
Watermark Detection System
Watermark or its presence
Key

be noted that there is no "global watermarking method." The work of Kutter and Hartung (2000) has divided watermarking application into four categories: watermarking for copyright protection, fingerprinting for traitor tracking, watermarking for copy protection and watermarking for image authentication.

Watermarking for Copyright Protection

The most vital application of watermarking today is the protection of one's intellectual property. The goal is to insert information about the source – the copyright owner – of the data in order to protect it from being claimed by others. Therefore, the purpose of watermarks is to establish rightful ownership. This application requires a high level of robustness. The focus of this application is the Web, which contains many images that the copyright owners wish to protect.

Fingerprinting for Traitor Tracking

Another type of application, "fingerprinting," is used to pass information about the legal recipient to identify single distributed copies of the data. This application requires the insertion of a different watermark into each distributed copy, a requirement that is helpful in tracing illegally produced copies of the data that may circulate. This method is equivalent to serial numbers in software products. Watermarks for fingerprinting applications require a high robustness against standard data processing, as well as attacks.

Watermarking for Copy Protection

The existence of a copy protection method to disallow unauthorized copying of media is a much-needed feature in a multimedia distribution system. Copy protection is not likely to be achieved in open systems; however, it is possible to use watermarks indicating the copy status of the data in closed systems. Consider DVD systems that embed copy information within the data as a watermark. The DVD player will contain copy control and copy protection mechanisms (Linnartz, 1998; Bloom et al., 1999) that use watermarking to signal the copy status of multimedia data, like "copy once" or "copy never."

Watermarking for Image Authentication

In an authentication application, the objective is to detect modification of the data, to be achieved with so-called "*fragile watermarks.*" Fragile watermarks are watermarks that are used in authentication applications in order to detect modifications of the data rather than conveying un-erasable information. Fragile watermarks have limited robustness.

ACTIVE WATERMARKING SYSTEM (AWS)

The Active Watermarking System (AWS) solves many of the concerns associated with the protection of digital intellectual property. The system addresses these concerns through the automatic insertion of hidden digital watermarks to establish copyright protection for ownership identification and fingerprinting for traitor tracking. In addition, the AWS maintains sufficient information to conduct digital data authentication, allowing content verification of a digital object utilizing the embedded watermark. The AWS basic functionality is supported by using various components, which include database tables, active rules, watermarking algorithms, and several user interfaces (Owner Registration, Owner Upload, Requestor Download, and Authenticate). Prior to discussing these components, the different AWS user roles, their responsibilities, and their actions are presented. These users consist of Object Owners (O_O) seeking copyright protection, Object Requestors (O_R) accessing the digital data and being fingerprinted, and Object Authenticators (O_A) that are determining digital data trustworthiness.

Initially, the O_O must register with the AWS through the Owner Registration interface, which generates and assigns a unique Owner Identification Number (O_{ID}) to each O_O. Subsequently, the AWS generates a unique Owner Watermark (O_{WM}) that is determined by the O_{ID} associated with each O_O. During O_O object submissions using the Owner Upload interface, each object is tagged with its owner's O_{WM}, thus protecting (copyrighting) the object. Therefore, each object is stored with a hidden O_{WM} ready for O_R rendering. Any changes to the owner's information will not affect the O_{WM}, which remains valid for all past and future object uploads.

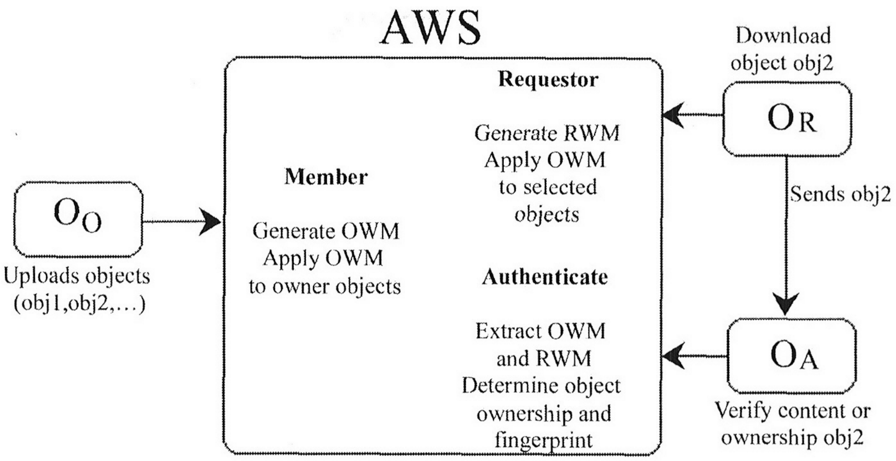
The O_R consists of Internet, intranet, and/or extranet users, based on the AWS deployment strategy. When the O_R accesses the AWS through the Requestor Download interface, the system generates a unique requestor watermark (R_{WM}). The R_{WM} is composed of the requestor's IP address and the time/date of the request; this information allows the system to track objects. Each object made available to the requestor must be tagged (fingerprinted) with the hidden R_{WM}. Based on the intended user population, the IP address is typically sufficient to identify a particular O_R, but for public Internet users the AWS must require the O_R to register with the system to obtain specific personal information.

The O_A receives an object and wants to determine the owner of the object or to verify the authenticity of the object. The O_A could be any user of the AWS that has obtained a copy of the object, either directly from the system as an O_R or indirectly from another O_R. Using the AWS Authenticate interface, an object is supplied to the AWS, which extracts the embedded O_{WM} and R_{WM} to determine the O_O, O_R, and/or a statistical degree of confidence on the object's content.

In Figure 2, we depict the AWS workflow, its various interfaces, and information inter-exchanged for the normal usage of the system. A significant feature of the AWS that is growing in importance in e-business is the ability to validate the contents of an object, as more digital data becomes a part of the business environment. Of concern in business is the possibility that a transmitted object could have been altered from the original object. In our system, an O_R obtains an object, changes its contents slightly, and sends it to a third party. Typically, it would be very difficult for the object's content to be verified for authenticity. Using the AWS, the object is verifiable using its O_{WM}, determining if any changes have been performed on the object. Once tampering is detected the AWS can provide the R_{WM} information that identifies the primary O_R.

When the O_O uploads an object, it is watermarked using the O_{WM} with an appropriate watermarking algorithm. The active component of our system automatically determines the watermarking algorithm based on the object's characteristics and O_{WM}. In this way, all objects stored within the context of the AWS are protected with the owner's watermark. In order to track object downloads, when an O_R accesses the AWS, a R_{WM} is produced in real time using information extracted from the cur-

Figure 2: AWS and its user interfaces



rent communication session. The active component in the AWS adds the R_{WM} to each object offered to the requestor in real time. The manner in which each object is fingerprinted with the R_{WM} is based on the object type, its characteristics, and the properties of the R_{WM}. An object is only made available for downloading or viewing if it has been augmented with both the O_{WM} and R_{WM}, uniquely identifying the owner and the requestor. The active rules in the AWS constitute the mechanism required to identify the object type, determine the object's characteristics, select a corresponding watermarking algorithm, and perform the watermarking off-line during object submission and in real time during object request. Further, the O_A can submit an object to the AWS, which will test the object's authenticity.

Database Structure

The basic functionality of the AWS utilizes three tables: the Member, Object, and Session tables. Using these tables we are able to provide the essential task of copyright, fingerprint, and authentication protection of an owner's digital property.

We are able to store a watermark for each O_O that the system maintains, an impossible task for the O_R as the number of object requestors and selected objects can be quite excessive. Data concerning the O_R and its computed R_{WM} is temporarily maintained in the Session table to avoid recalculation during the current communications session. Although the R_{WM} is not stored in the AWS beyond the current session, it is embedded for each O_R rendered object. The Member table stores O_O's registration information, which minimally consists of a record with an O_{ID}, the member's name, address, and affiliation. Inserting an owner record into the Member table triggers the Generate_OWM rule, which appends an AWS computed watermark to the member's information prior to adding the record to the table.

The objects submitted by AWS members are placed in the Object table. An Object table record consists of the owner's O_{ID} and the O_{WM} watermarked object. When an object is submitted, the Save_Object rule is triggered, which, according to the object's type and characteristics, selects the most suitable watermarking algorithm, and applies the owner's O_{WM} to the object. A second rule associated with the Object table is the Request_Object rule, which is triggered in response to the retrieval objects, applying a generated R_{WM} to each object supplied to the particular requestor. To perform necessary operations of an O_A, a set of DBMS stored procedures and functions are required. These stored modules process a submitted object for authentication, by extracting its O_{WM}, and subsequently searching the Member table for a matching O_{WM}, identifying its O_{ID} leading to the original object call for further processing.

Active Rules

The four basic rules below comprise the core of our data protection system. These rules take the form of ECA rules, which are supported in many commercial DBMS. Although not shown, there exist various versions of rules 2 and 4 in the system that handles the necessary watermarking task. These rules check the type of an object and its characteristics before executing a specific corresponding procedure and algorithm. For example, an object inserted into the Object table would trigger all rules associated with this event. The condition part of each rule would check the object, ultimately identifying a single rule from the triggered set to be used and processing the object with the most effective watermarking approach. Rules 1 and 3 are responsible for augmenting inserted record data with a system-generated watermark. These rules are responsible for verifying the uniqueness of the computed watermark in order to guarantee distinct object ownership.

Watermarking Techniques

The AWS is a protection scheme that provides reliable methods for efficiently watermarking an object and that authenticates a watermarked object. The use of any of the methods of watermarking is application-dependent. The design of AWS does not focus on supporting a single watermarking technique; however, it is flexible so that any watermarking method can be used.

Watermarking techniques have emerged as the leading solution of ownership and content authentication for digital media documents. Watermarking algo-

Rule No: Rule Name	Event	Condition	Action
Rule 1: Generate_OWM	Inserts member data into Member table	Is member data unique	Processes member data and generates O_{WM} using O_{ID} to store along with the member data
Rule 2: Save_Object	Inserts object into Object table	Is it an Image with dimensions less than 640x480	Processes Image with corresponding watermarking algorithm with member's O_{WM}
Rule 3: Generate_RWM	Inserts requestor data into Session table	Is requestor data unique, obtain from communication link	Processes requestor data, generates and stores temporarily in the Session table the requestor's R_{WM}
Rule 4: Request_Object	Selects objects from the Object table	Is it an Image with dimensions less than 640x480	Obtains the requestor's R_{WM} from the Session table and using a corresponding watermarking algorithm tags each object

rithms must address the following issues:

- Ratio between the information contained in the watermark and in the host signal (image, video, audio, etc.)
- Image degradation due to watermarking
- Robustness of the watermark to transmission distortion of the image

The ultimate watermarking methods should resist any kind of distortion introduced by standard or malicious data processing. No perfect method has been developed yet; thus, practical systems must implement a compromise between robustness and the competing requirements such as invisibility and information rate. For example, in image watermarking, if we need a method that is resilient to JPEG compression with high compression factors, it is probably more efficient to employ a method that works in a transform domain rather than a spatial domain (Kutter and Petitcolas, 1999).

The watermarking algorithm consists of three stages: generating, embedding and detecting. The generating stage is an off-line process (a process that is not performed in real time). There are two embedding stages: embedding the author's watermark, which is an off-line process, and labeling the requestor information, which is an on-line process (performed in real time). The most crucial stage is the detection stage, an on-line process. The detection technique is applied to a large set of images; therefore, a fast and efficient detection method is desirable. AWS adopts the detection algorithm D:

$$D(S,K,O_{WMi}) = \begin{cases} 1 \text{ if } W_E \approx O_{WMi} \\ 0 \text{ otherwise} \end{cases}$$

where S is the submitted object, K is the AWS key used to enforce security, W_E is the extracted watermark and OWMi is the watermark for owner *i*. The relation H• indicates that A is similar to B. W_E H• O_{WMi} indicates that W_E is equal to O_{WMi} or some confidence measure indicating how likely it is for the given watermark O_{WMi} to be present in S.

AWS IMPLEMENTATION

The AWS is a three-tier Web application utilizing various technologies at the respective processing sites–client-side, Web-server-side, and data source. On the client-side, there are static and dynamic Web pages comprising the various AWS user interfaces necessary in obtaining and viewing objects. Objects are uploaded using a form's "put" option and displayed by inserting the binary data comprising the object (image) into the Web pages. At the Web-server-side, Microsoft's Internet Information Server (IIS) provides the basic application logic to interact with an Oracle 9i enterprise database. At the data source, Oracle's PL/SQL language is employed to implement the system's triggers, which invoke calls to stored procedures and functions programmed in the Java programming language. These triggers and database executions consist of all processing necessary for object watermarking and authentication. Therefore, the AWS is execution-intensive at the ADBMS, while being less demanding at the Web server and browser levels.

The storage and manipulation of binary large objects (BLOBs) in Oracle and Java are required to handle the digital data objects. Oracle is an object relational database with active rule facilities, which allows the definition of user-defined data types that encapsulate attributes as well as behaviors. The user-defined Entity_Type contains a BLOB attribute named "item" used to hold the binary data in the database. The item attribute cannot be directly selected from the database through queries (though one can query the length of the item attribute). Consideration was given to alternative methods such as using tables of predefined Oracle data types to store BLOBs. These are easier to develop but do not apply the object-oriented concept that is sought for robustness and program logic. The Object table is created containing a column data of Entity_Type, which maintains the BLOB, other attributes, and several manipulation methods. The Oracle JDeveloper 9i was the selected tool used to map the Oracle user-defined type Entity_Type to the Java class. Once mapped to a Java class, the BLOB can be accessed and manipulated using Java code to process the object. These BLOBs are passed to the VB.net code as an intrinsic BLOB type for rendering at the client-side.

While some research exists on watermarking video and audio, the majority of publications in the field of watermarking currently address the copyright of still images. Without significant loss of generality, we focus on watermarking still images. Therefore, the initial AWS implementation handles PGM images but can be expanded to other digital media with the incorporation of additional water-marking algorithms and the development of supporting Java classes through the use of inheritance.

AWS PERFORMANCE EVALUATION

The AWS includes several key capabilities and features that have been described by researchers in the field of electronic copyright management systems. These features include the ability of the AWS to detect modification to copyrighted properties and the ability to identify the requestor of the materials and maintain records of users and their copyrighted materials. The AWS accomplishes these primary activities through processing conducted at the database shared among all AWS users. Therefore, performance analysis of the AWS is necessary to evaluate the cost of the watermarking process, the performance of the AWS under different loads, and the extra delays imposed on AWS users.

In order to determine the execution cost associated with watermarking and to obtain a baseline value to compare the operations of R_{WM}, images of various sizes (50K, 150K, 250K, and 1000K) are retrieved without watermarks. Obtaining these values provides a reference to establish the percentage increase in execution time associated with watermarking. We focus on the embedding of R_{WM} and the detection of the watermark because they are real-time processes that significantly impact the system's performance, as opposed to examining the insertion of images and tagging them with an $O_{WM.}$ The percentage increases associated with the R_{WM} images is attributed to the cumulative time required to generate/store the R_{WM}, identify a trigger, and watermark the image. The results in Table 1 indicate the percentage difference when a watermarked image is retrieved compared to its non-watermarked form. For example, relative to an original image, it would take 7.89% more time to watermark and retrieve an image of size 250K. The disparity among percentage increases changes according to the image size, since the watermarking algorithm must process a larger image. This percentage increase highlights the cost of watermarking, as trigger activation and R_{WM} production remain consistent across the various image sizes.

The previous results focus on a single AWS user; in order to estimate the performance of the AWS under different loads, simultaneous object selects are performed to ascertain the system's response time compared to that for a single user. Table 2 contains the time increases of multiple users relative to a single user. The AWS being database-processing intensive does extend the system's response time according to the number of users (which supports our future research consisting of a network of AWS to offload and improve system performance). This is apparent from the higher values in the table, which indicate a user's mounting delay as numerous simultaneous transactions are taking place, extending the resources of a single AWS installation. A user that accesses the AWS as the 30th active user would suffer a postponement of 26.4 times a single user in retrieving an image of size 250K.

An extra delay associated with the detection process during image authentication can be attributed to the identification of the original image. Currently, an extracted O_{WM} from a submitted image is used to find a match in the Member Table leading to the original image to conduct the process. As the number of O_O increases, the O_A performance degrades, because there exists a corresponding increase in stored O_{WM} to match. A solution to this problem is to use a watermarked image histogram. The detection process changes in these ways:

Table 1: User percent increase

Image size	50K	150K	250K	100K
Increase time	5.49%	6.90%	7.89%	11.02%

Table 2: Number of simultaneous requests and image sizes

Image Size	10 users	30 users	50 users	70 users
50K	3.59	5.78	8.4	17.5
150K	4.56	7.3	10.9	30.3
250K	10.21	26.4	39.8	57.3
1000K	16.23	32.8	45.8	60.2

- When the Object Authenticator submits an image I, the histogram H_{OA} of that image is generated.
- The histogram H_i of the watermarked images stored in the database DB are generated and stored along with the images in the database. The matching algorithm M is performed to determine whether the supplied image has been watermarked by AWS.

$$M(H_{OA}, DB) = \begin{cases} 1 \text{ if } H_i \in DB \text{ and } H_i \approx H_{OA} \\ 0 \text{ otherwise} \end{cases}$$

The relation H• indicates perceptual similarity between the two histograms. H_i may not be equal to H_{OA} because of the R_{WM} embedding process, which tags an image with the IP address or personal information of an O_R. The use of the matching algorithm M and a multidimensional index, formed from the image's vector histogram, would reduce the number of images to consider. This would improve the authentication process as a result of increasing the storage capacity of the implementation and reduce the time required to detect a watermark in an image and related information. The addition of this capability forms part of our future work to improve the AWS performance.

FUTURE WORK

The proposed system can target users associated with an intranet, extranet, and/or Internet population. An intranet deployment would consist of using the AWS within the boundaries of an organization; while an extranet deployment would extend the user base to consist of corporate partners that have access to an organization's information. In both scenarios, objects can be registered, obtained, and authenticated for a limited set of users, augmenting an organization's object processing approach beyond an object repository. This deployment provides a level of data protection that is often necessary in business-to-business transactions in order to remove any suspicion of improprieties. A single centralized AWS installation is sufficient to protect data within these e-business contexts, possibly housing the system at the organization's central office. Expanding the systems outside these constraints requires various deployments of the AWS to handle widespread use from among the Internet population. This expansion requires a network of AWS whose systems communicate among themselves when a member registers or an object is authenticated. When a new member attempts to register with the systems, it is no longer valid to generate a unique watermark from the registering AWS; instead, it must be a universal AWS watermark unique throughout all AWS installations. In addition, when an object is authenticated at an AWS, it must be checked at all AWS that comprise the AWS network. This is an area of ongoing research requiring additional active components, AWS communications protocols, and expanded fingerprinting capabilities.

CONCLUSION

The digital world has brought about new protection requirements for proprietary information and data for businesses, individuals, owners, and creators of such valuable items. The ability to protect and authenticate the ownership of these electronic items will encourage an increase in e-business and enhance the Internet. The AWS proposed in this paper addresses these issues of ownership and authentication, combining the technologies of watermarking and active database to establish the necessary protection requirements. The combination of these technologies establishes a powerful method for marking digital media to identify ownership and maintain data integrity and avoid potential misuse of the media. Furthermore, the AWS fingerprints requested media with information associated with the requestor, which allows the system to determine when the media has been modified and who originally obtained the media.

To date, the widespread use of watermarking as a tool has not been fully exploited in business. These systems are often developed in response to the unauthorized misuse of copyrighted materials over the web (especially record labels and publishing companies). Although many watermark embedding and recovery systems are readily available, the standard is to develop custom-built applications that are specific to a watermarking technique. As the AWS demonstrates, the possibilities for the widespread use of more general applications utilizing watermarking of digital media are significant. The AWS has much to offer in protecting the intellectual and creative property of individuals and organizations in the digital age, while providing a flexible and scalable system that rapidly incorporates and manages new media types.

REFERENCES

Bloom, J. et al. (1999). Copy Protection for DVD Video. *Proceedings of the*

IEEE, 87(7), 1267-1276.

Burns, C. Inc., (1996). *Copyright Management and the NII: Report to the Enabling Technologies Committee of the Association of American Publishers.* Washington DC: Association of American Publishers.

Katzenbeisser, S. and Petitcolas, F. (2000). *Information Hiding: Techniques for steganography and digital watermarking.* Artech House Books.

Kutter, M. and Hartung, F. (2000). Introduction to watermarking techniques. In Katzenbeisser, S. and Petitcolas, F., editors, *Information Hiding: Techniques for Steganography and Digital Watermarking.* Artech House Books. pp. 97-120.

Kutter M. and Petitcolas, F. (1999). Fair Benchmarking for Image Watermarking Systems. *Proceedings of Electronic Imaging '99 Security and Watermarking of Multimedia Contents.* 3657, 226-239.

Linnartz, J.P. (1998). The 'Ticket' Concept for Copy Control Based on Embedded Signaling. *Proceedings of the 5th European Symposium on Research in Computer Security.* 1485, 257-274.

Pons, A. and Aljifri, H. (2002). An Active Watermarking System. *Issues in Information Systems*. 3, 515-521.

Stefik, M. (1996). *Internet Dreams; Archetypes, Myths and Metaphors.* Cambridge, Massachusetts, USA: MIT Press.

Stefik, M. (1997). Shifting the Possible; How Digital Property Rights Challenge Us to Rethink Digital Publishing. *Berkeley Technology Law Journal,* 12, 137-159.

Smith, K. and Webber, F. (1995). A New Set of Rules for Information Commerce-Rights-Protection Technologies and Personalized Information Commerce Will Affect All Knowledge Workers. *Commercial Week,* November.

Voyatzis, G. and Pitas, I. (1999). Protecting Digital Image Copyright: A Framework. *IEEE Journal Computer Graphics and Application,* 19(1), 18-24.

Widom, J. and Ceri, B. (1996). *Introduction to Active Database Systems*. San Francisco: Morgan Kaufmann Publishers, Inc.

World Intellectual Property Organization (WIPO) (1996a). *Diplomatic Conference on Certain Copyright and Neighboring Rights Questions*. WIPO: Geneva.

World Intellectual Property Organization (WIPO) (1996b) *WIPO Copyright Treaty.* WIPO: Geneva.

Alexander Pons is an assistant professor in the Computer Information Systems department at the University of Miami. He received his Ph.D. from the University of Miami in Electrical and Computer Engineering in 1998. Dr. Pons has over fifteen years of industry and academic experience as an engineer, consultant, and professor. For the past several years, he has been involved in various aspects of real-time systems and databases as a researcher and developer. He has published in several international journals and conferences. His research interest includes real-time systems, programming languages, databases and Internet technology.

Hassan Aljifri is an assistant professor in the Computer Information Systems department at the University of Miami. His research interest includes computer and network security, programming languages and Internet technologies. He has published in several international journals and conferences. Most of Dr. Aljifri's recent research has been focused on methodology, framework, and techniques for designing secure systems.