TRACING DATA DIFFUSION IN INDUSTRIAL RESEARCH WITH ROBUST WATERMARKING

Christoph Busch and Stephen D. Wolthusen
Security Technology Department
Fraunhofer-IGD
Darmstadt, Germany.

Abstract - This paper presents a security system for enforcing security policies throughout distributed environments. The aspects of the system dealing with the protection of digital data using object labeling and mandatory encryption at the OS level are covered briefly; the main focus is on protection provided in the analog domain. This is accomplished by embedding multiple watermarks identifying the copyright owner, the identity of the object, and of users accessing the object into any markable object accessed by users.

INTRODUCTION

Protection of trade secrets in industrial R&D is an important factor in a competitive global environment with limited chances of recourse to legal protection against industrial espionage by either competitors or national intelligence services in the support of their respective national industries. While defense against external adversaries are required, protection against insider attacks or simple carelessness is also required; this has been the indication of, among others, the annual computer security surveys commissioned by the FBI [7]. Security policies covering data diffusion and deliberate interception of confidential material (e.g. design drawings) due to internal risks are therefore being set up in several industrial sectors such as e.g. mechanical engineering, semiconductor industry, as well as in the automotive and chemical/pharmaceutical sectors. Digital watermarks play an important role in the enforcement of such security policies if these are to cover not only the digital but also analog work-flow. The latter is particularly important since the automatic safeguards available in the digital domain do not exist for analog representations; instead, one is usually confronted with blatant disregard of fundamental security procedures in the name of simplicity, efficiency, and convenience. In order to trace back the origin of document leakage, a robust watermark must be detactable from hard copies created in the secured area. This can be accomplished by embedding a watermarking component in the operating system, thus providing an application-independent mechanism that will embed digital watermarks regardless of the application and, in addition, is impervious to user manipulation. The requirements imposed on an effective algorithm for use in such a mechanism are:

- The algorithm must allow multiple non-interfering digital watermarks
- The watermarks must be robust against D/A conversions (e.g. printing)
- Algorithm performance must permit embedding in real time
- Blind detection schemes must be used
- Visibility of the watermark should be low to imperceptible.

Adversaries are not expected to mount even moderately elaborate attacks such as StirMark [11], collusion attacks [2], or copy attacks [9] in this industrial setting. This is due to the following assumptions:

- The attacker does not have full control over the working environment. This implies that he will not necessarily have the tools required to conduct the above-mentioned attacks at his disposal.
- An attacker wishing to surreptitiously acquire information from a source will ensure that the original, potentially compromising material is not exposed to third parties that could identify the source and thus expose sources and methods.

IPR PROTECTION SYSTEM

The goal of the ongoing CIPRESS project (Cryptographic Intellectual Property Rights Enforcement SyStem) is the provision of a comprehensive security architecture that can be retrofitted onto existing, particularly COTS (commercial off the shelf) operating systems. This security architecture is designed to provide protection in digital and analog domains. In the digital domain this is accomplished by inserting interception points into the host operating system at all junctions where interactions with the outside world occur, e.g. file system and network interfaces. These points allow the introduction of access and use control that is enforced by a centralized security policy mechanism. An attacker will always pursue the avenue of least resistance — data in human-readable, i.e. analog form. Unless one is willing to completely disrupt the normal workflow, it is difficult to prevent operations such as printing for regular office work. An attacker can thus either print out whatever information he desisres or steal the relevant data in the form of someone else's printouts. In cases where sensitive material is discarded without prior shredding or burning, this is not even considered theft in many jurisdictions. Similar considerations also apply to other analog output mechanisms such as audio I/O jacks. While using devices for recording audio output is somewhat more conspicuous than paper printouts, the amount of data that can be transmitted is considerable. This applies to both original audio material [1] and audio material containing steganographic signals. In summary, any security system that does not deal with the problem of analog representations leaves potentially significant problems unresolved[2].

Object Registration Before discussing the watermarking aspects of the CIPRESS project, a brief exposition on the data object model is required. CIPRESS distinguishes three types of data objects. Category I (C-I) data objects are plaintext data objects, typically external data objects accessed using a read-only mechanism or are required to start up the host OS up to the point where the security extensions are loaded. Any

[1] e.g. intelligence material gathered by leaving a portable computer's microphone activated during confidential meetings; one can also use a suitably equipped PDA as a recording device for later uploading, both avoiding conspicuous RF emanations while their use as a recording device can be plausibly denied

[2] Beyond this there are obviously issues with shielding data processing equipment against direct analysis of their emanations, these are beyond the scope of this paper.

file accessed by a user[3] with write access permitted is automatically encrypted using a key specific to the machine where the access originated and thus becomes a Category II (C-II) data object. This implies that exchanging C-II data objects across node or user boundaries is not possible. Exchange of data objects is restricted to Category III (C-III) (including derived C-I) objects. These objects are required to be registered with a central instance, the Key Center (KC). As a result of this process, any and all subsequent use of a registered data object is tracked by the KC. This is possible since an object label is affixed to each data object, containing an unique identifier; this label is handled by the OS extension and never revealed to applications. CIPRESS protects the data in transit and when not running by mandatorily encrypting the data objects at all times except for temporary in-memory representation. As keys are used inside the (ideally tamper-resistant) OS extension only and discarded immediately after use, each initial access or use leading to an in-memory representation leads to an access and use control verification and implicitly generates an audit event. C-III data objects are also integrity-protected when stored on client systems. Any modification to such a object (document) leads to a change in status to a C-II document; a server maintains a copy of each registered object which is also protected with a digital signature provided by the originator (this signature cannot be verified on client systems due to the effects of watermarking). This mechanism allows changing the access and use control policy for an object once at the central KC, resulting in the immediate application of these rules to any instance of the object regardless of its storage location on either servers or clients. Actual registration is performed through intermediate servers called Content Servers (CS). These are grouped into organizational units and provide a trusted environment for a given domain.

SERVER-SIDE WATERMARKING

CIPRESS assumes the availability of a multi-watermark capable system in its semantics so as to permit the use of several digital watermarks for different application purposes with a payload of 64 bits for each hierarchy level. During the registration process discussed below, a C-I or C-II data object is uploaded to one of the CIPRESS archiving facilities. These act as data repositories and provide services such as storage of meta-data associated with the data objects. Registration consists of creating an unique label for the object derived from the data (typically plaintext data transmitted over an integrated VPN channel) and assignment of the required access and use control rules. In cases where the data to be registered is detected to be of a type that can be marked[4], the CS will apply two watermarks to the object to be registered.

Owner Watermark The first watermark applied is a secret marking, i.e. it is assumed that the key for retrieval is known only to the KC, trusted by default, and to the operator of the CS, also a trusted role. The actual payload to be embedded is a fixed but arbitrary bit string. The selection of the payload and its timestamp at the server side protects against inversion attacks. It is therefore possible to retrieve only a fraction of

[3] or, depending on the host OS in which CIPRESS is embedded, even touched by the system itself

[4] i.e. not already superencrypted and of both a media type and format that is recognized by one of the media type modules registered

a payload and provide a probability for a match provided a given retrieval key; this is relevant in cases where the watermark recovery is not fully successful due to e.g. signal degradation or deliberate attacks on the digital watermark.

Retrieval Watermark The retrieval watermark is embedded using the same algorithm; the key used here is assumed to be public. While permitting deletion and overwriting attacks, this threat is not relevant in this case since the purpose is to provide an additional benefit, i.e. retrieval of the original, integrity-protected, authenticated document stored in an archive server given only an analog copy (e.g. a fragment of a printout). The payload consists of two parts. One is a numerical ID of a CS archiving the object (16 bits); the other is derived from the original object as follows: The original object is processed using a cryptographic hash function (e.g. SHA-1). The second part of the payload then consists of 48 bits selected arbitrarily from the hash[5]. Which bits are selected is irrelevant due to properties of the algorithm. Given an analog representation, one can obtain the retrieval watermark, present the hash fragment to the CS identified in the same payload. It is still possible to retrieve the proper data object[6].

Supported Data Types and Key Considerations Watermarking in CIPRESS is designed to be modular and agnostic with regard to the algorithms used. The only requirements are those given earlier. Each data object is presented to a sequence of recognizers claiming data types and representations and transforming data in situ, if necessary signaling for storage extension. Data types for which our group[7] has developed algorithms are still images, video (including AVI and MPEG-2 broadcast quality) [5], audio [1], and three-dimensional polygonal models [4]; other modules such as watermarks for formatted text are easily embedded in this architecture. The initial prototype contains a module for still images. The algorithm used for this is a variant of [3] and operates on a meta-format for which various filters exist. The algorithm selected here was chosen since it fulfills the real-time constraints for still image and video data as well as those given earlier. The selection of the secret keys for embedding must balance the need of protection against collusion attacks against the time required to test the watermarks in case retrieval is necessary. We therefore restrict ourselves to embedding user fingerprints with user-specific keys in all data objects the user touches. This still opens possible attacks for colluding users but makes blind detection with large lists of suspects feasible.

Detection of Markable Data Objects CIPRESS as presented here operates on finite data objects only, i.e. data objects which can be represented in a file. A data object to be converted at the CS into a C-III data object is uploaded as a file given a format which allows identification of the start of a file. The design assumption is that supported file formats contain a "magic number" sufficient to trigger additional mark-

[5] This is required since watermark payload is limited; we have chosen to use an uniform 64 bits of payload size although that limitation is arbitrary and can be increased at the price of either increased fragility or perceptibility provided the algorithms used support such variable payloads to begin with.

[6] Due to avalanche properties of the hash algorithm, this is true even in the case of a hash collision in the 48 bits presented to the server providing a single collision probability of 10^{-12}

[7] See `http://syscop.igd.fhg.de`

ability decision steps. Each supported media type module (MTM) must register itself with the detector dispatcher (DD) and provide a list of matchings to the DD. The DD matches magic numbers sequentially as per the sequence of MTM registrations, calling the matching module which may cause additional decoding steps during which the MTM can still decline to mark the file. The DD is oblivious to the marking mechanism. Detection of media types at client nodes uses the same DD mechanism but requires additional pre-processing steps to obtain the magic numbers. If a C-III file is read from the file system and decryption is successful, the full object is forwarded to the DD which then proceeds as above. The second ingress path covered by the DD is for network-originating objects. A filtering mechanism residing in the network stack uses a sliding window to detect this object label, detects and temporarily withholds data as seen above (for additional details on this please refer to [12]).

EMBEDDING FINGERPRINT WATERMARKS IN THE OPERATING SYSTEM

Besides the proof of ownership provided by the owner watermark described in section , CIPRESS also embeds digital watermarks on the individual clients working with C-III data objects. As shown in figure 1, CIPRESS does this by funneling each access to a data object, regardless of its origin (network, file system) through a central facility where a labeled object is identified, the access permissions including key material retrieved from the KC, the data object decrypted and then forwarded to the watermarking subsystem. As described in section , the data object is matched to a type and representation that can be watermarked. What is embedded then is a watermark containing the identity of the entity (i.e. typically an individual) under whose authority the data object was accessed as the payload; this is embedded using a secret key also obtained from the KC. As this is a computationally expensive operation, the operating system extension tries to keep track of the marked object and work from cached data as long as possible. In case the object is written by the user/application program, this occurs with the watermark embedded; the embedding mechanism ensures that the same user identity is embedded only once. As a result each markable object contains the user's fingerprint regardless of the application used to view, print, or otherwise process the data object.

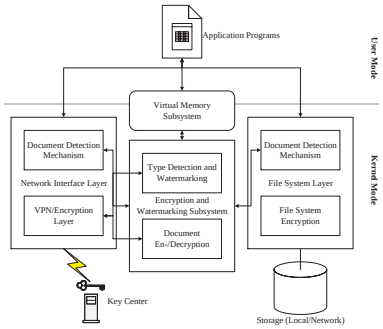

Figure 1: Integration of Watermark Components into the Operating System

CONCLUSIONS AND OUTLOOK

The CIPRESS project (additional details on which can be found in [6], see also [8]) is currently in the process of commercial exploitation as the ReEncryption System by the Mitsubishi Corporation. The presence of digital watermarking and hence the rather unique ability to protect both digital and analog representations has been one of

the core reasons for the commercial acceptance of the system in areas requiring very high security levels.

ACKNOWLEDGMENTS

The authors would like to thank the Mitsubishi Corporation, Tokyo, Japan, for excellent cooperation and support in realizing the system described here.

References

[1] ARNOLD, M. Audio Watermarking: Features, Applications and Algorithms. In *Proceedings of the 2000 IEEE International Conference on Multimedia* (July 2000), IEEE, IEEE Press.

[2] BONEH, D., AND SHAW, J. Collusion-secure fingerprinting for digital data. In *Proc. 15th International Advances in Cryptology Conference – CRYPTO '95* (1995), pp. 452–465.

[3] BURGETT, S., KOCH, E., AND ZHAO, J. Copyright Labeling of Digitized Image Data. *IEEE Communications Magazine 36*, 3 (Mar. 1998), 94–100.

[4] BUSCH, C., AND BENEDENS, O. Towards Blind Detection of Robust Watermarks in Polygonal Models. In *Proceedings of EUROGRAPHICS 2000* (Aug. 2000), European Association for Computer Graphics.

[5] BUSCH, C., FUNK, W., AND WOLTHUSEN, S. Digital watermarking: From concepts to real-time video applications. *IEEE Computer Graphics and Applications 19*, 1 (Jan./Feb. 1999), 25–35.

[6] BUSCH, C., GRAF, F., WOLTHUSEN, S., AND ZEIDLER, A. A system for intellectual property protection. In *Proceedings of the SCI 2000/ISAS 2000, Orlando, FL* (July 2000), pp. 225–230.

[7] CSI. 6th Annual Computer Crime and Security Survey. Tech. rep., Computer Security Institute, San Francisco, CA, Mar. 2001. Commissioned by the FBI.

[8] KOCH, E., SAITO, M., AND ZHAO, J. U.S. Patent US6141753: Secure distribution of digital representations, Oct. 2000. Granted October 31st, 2000.

[9] KUTTER, M., VOLOSHYNOVSKIY, S., AND HERRIGEL, A. The Watermark Copy Attack. In *Electronic Imaging 2000, Security and Watermarking of Multimedia Content II* (Jan. 2000), vol. 3971, IS&T and SPIE.

[10] LOW, S. H., MAXEMCHUK, N. F., BRASSIL, J. T., AND O'GORMAN, L. Document Marking and Identification using Both Line and Word Shifting. In *Proc. IEEE INFOCOM '95, pp. 853–860.*

[11] PETITCOLAS, F., AND KUHN, M. *Watermark Robustness Testing Software.*

[12] RADEMER, E., AND WOLTHUSEN, S. *Transparent Access To Encrypted Data Using Operating System Network Stack Extensions. In* Communications and Multimedia Security Issues of the New Century: Proc. CMS'01 *(May 2001), R. Steinmetz, Ed., Kluwer, pp. 213–226.*