DIGITAL RIGHTS MANAGEMENT: THE TECHNOLOGY BEHIND THE HYPE

Mark Stamp
Computer Science Department
San Jose State University
stamp@cs.sjsu.edu

ABSTRACT

A specific digital rights management (DRM) system designed for digital document protection is presented in some detail. Following some background information on document protection, the various technical measures employed by this DRM system are discussed. These technical measures include a variety of tamper-resistance methods, controlled execution techniques, encryption, digital watermarking and other techniques. The discussion of specific security features is followed by a discussion of several other current and developing DRM systems and technologies. Comments and conjectures on the limitations and future directions of the technology are made throughout.

1. Introduction

Digital rights management (DRM) can be viewed as an attempt to provide "remote control" of digital content. The required level of protection goes beyond simply delivering the digital content—restrictions on the use of the content must be maintained after it has been delivered. In other words, DRM requires "persistent protection", i.e., protection that stays with the content.

For example, consider a digital book. Such a book can be delivered over the Internet using standard cryptographic techniques. But if the recipient can save the book in an unrestricted form, he or she can then freely redistribute a perfect digital copy to a large percentage of the population of the world. This fear has led publishers to largely forego the potentially lucrative sale of digital books and has had a similar chilling effect on the legitimate distribution of other types of digital content. Without robust DRM, owners of digital content have little choice other than to rely on the honor system. (See Sayer, 2000, for a discussion of Stephen King's less than successful online experiment with the honor system.)

There is a mature and robust cryptographic theory that can be applied to the problem of securely delivering digital content. Unfortunately, there is no comparable theory currently available for the DRM problem.

An effective DRM method would have far-reaching implications. For example, armed with strong DRM, an individual could maintain control over online personal data. In fact, it has been conjectured that online privacy can only survive if DRM succeeds (Geer, 2002). However, the general consensus seems to be that such high-level security is unobtainable via DRM, at least in the current PC-dominated world (Cryptographers Panel, 2002). In addition, recent research supports a pessimistic view of software obfuscation (Barak et al., 2001), which has been touted as a potent DRM-enabling technology (Collberg, Thomborson, and Low, 2000).

The DRM market has been estimated to be worth $3.5 billion by 2005 (PDFzone.com, 2001; Rosenblatt, Trippe, and Mooney, 2001). Of course, such projections must be treated with suspicion. Nevertheless, it is clear that there is a large potential market due not only to the obvious concerns of copyright holders, but also to laws mandating increased security for private information held in digital form (HIPPA HQ, 2003). DRM within enterprises is another huge potential growth area, which, due to the attention focused on copyright issues, is often overlooked. Not surprisingly, the DRM market is a crowded place. There are literally dozens of active DRM companies, along with a growing number of formerly active companies.

Most marketers of DRM products state—or at least strongly imply—that their proprietary DRM solution can provide unbreakable protection from unauthorized use. Even the opponents of digital content protection often talk in dark tones about the consequences of DRM which, they argue, would allow copyright holders to enforce excessive restrictions on "fair use" (von Lohmann, 2002).

However, several DRM products that have been widely available—such as those from Adobe (Guignard, 2003; Bailey, 2001) and Microsoft (Beale Screamer, 2001)—have been spectacular failures. Given this unimpressive track record, it is worth asking, what is the technology behind the DRM hype?

DRM products can, roughly speaking, aim for one of the following four distinct security levels. At the bottom rung are systems that rely on the honor system. In these systems, no real attempt is made at technical security

protection. Instead, such systems rely on users (or programmers) to "do the right thing". This model is somewhat analogous to the shareware distribution of software. Not surprisingly, these glorified honor systems have had limited success in the marketplace. At a slightly higher level are systems that employ an extremely limited, software-based, technical means of protection. For example, such a system might attempt to protect PDF documents by simply disabling the "save as" feature in the Acrobat Reader. These systems can only hope to deter the most naive users. A user who is knowledgeable enough to operate a screen capture program is likely to be able to defeat such a system.

A very few software-based DRM systems aim for a higher level of security. Such systems try to attain a measure of "controlled execution", where the user cannot easily do certain operations (e.g., screen capture) that might compromise security. In addition, such systems employ tamper checking techniques, controlled rendering, and various other methods. At the highest security level are systems that rely on tamper-resistant hardware. This is not yet a realistic option for systems intended for personal computers or similar devices. These four levels of protection are summarized in the table below, with an example system given for each level.

Security Level	Example
Honor system	National Academies Press (Casti, 2003)
Minimal software-based protection	Adobe eBooks
Maximal software-based protection	MediaSnap
Tamper-resistant hardware	"Trusted computing" (Anderson, 2003)

This paper discusses—at some length—a DRM product from MediaSnap, Inc. This system clearly aims for the highest level of software-based security. Since this information is taken from an actual product, specific details that might compromise its security cannot be disclosed. Instead, this system is described at a level of detail consistent with a semi-open design, as advocated by Anderson (2001). In the DRM marketplace, closed designs are clearly the norm. Consequently, the completeness and level of detail provided in this paper is far beyond that which is publicly available on many other functioning DRM systems. The MediaSnap system presented in this paper provides a reasonable yardstick against which other comparable DRM systems can be measured.

The paper then discusses several current or recent DRM offerings. Publicly available information about these systems is limited and based mostly on generic "white papers" designed more for marketing purposes than for technical enlightenment. Nevertheless, it is possible to infer certain aspects of the technology and it is instructive to compare the different approaches taken by various companies. Finally, the paper briefly returns to the topic of tamper-resistant hardware and points out some of the paradoxical security implications of such systems.

Before delving into the MediaSnap system, the paper first digresses to explain why standard methods of encryption—regardless of how well they are implemented—are insufficient for the level of protection required in DRM. In subsequent sections it will become clear that many DRM companies have yet to grasp this fundamental concept.

2. Encryption: Necessary but not Suffcient

Consider the following classic scenario. General G wants to communicate with Lieutenant L, where L is in the field with the troops and G is comfortably situated at headquarters. Suppose the two parties have a pre-determined symmetric key available (if not, the first step would be a key exchange using public key cryptography). General G uses his crypto-algorithm with the specified key to encrypt his message to L. The resulting ciphertext is then transmitted to L. Upon receiving the encrypted message (i.e., ciphertext), L decrypts the message using the known crypto-algorithm and the same key that was employed by G. In this scenario, which is illustrated in Figure 1, an attacker only has access to the encrypted message and only when it is transmitted from G to L. Consequently, an attacker must attempt to deduce the plaintext from the captured ciphertext (a process that is made much easier if a spy such as John Walker Jr. happens to work for G; see Earley, 2003).

Cryptography was designed to make the recovery of plaintext from the ciphertext (and, perhaps, other available information, such as limited amounts of plaintext) computationally infeasible. Carefully implemented cryptography effectively solves this problem.

Now suppose that Lieutenant L—along with his cryptographic equipment and keys—is captured by the enemy. In this case, the entire dashed box in Figure 1 (if not more) is in the hands of the attacker. This is analogous to the DRM scenario, where we are attempting to restrict the actions of the intended recipient. Clearly, cryptography was not designed to solve this problem. Therefore, other means of protection must be employed.

Given that cryptography is insufficient protection, is it necessary? Correctly implemented strong encryption assures us that converting ciphertext to plaintext without access to the key is computationally infeasible. Therefore, it is necessary that DRM employ strong encryption in order to eliminate the possibility that an attacker can remove

the protection without first recovering the key. But, again, encryption alone is not sufficient to provide persistent protection. At a minimum, the encryption key must be protected, which presents a tremendous challenge on an open architecture such as a modern personal computer. Since an attacker can recover a crypto key by reverse engineering the software that contains (or accesses) the key, in a DRM system it is necessary that the reverse engineering problem be as difficult as possible.

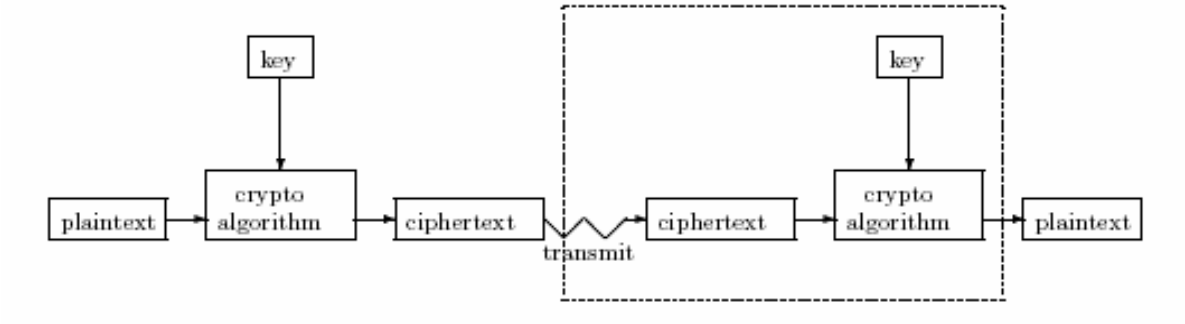

Figure 1: Cryptography

3. Overview of the Current State of DRM

On what does current DRM security rest? The theme of this paper is that the strongest security measures currently available for DRM rely on "security by obscurity" as opposed to any sound theoretical basis. Worse yet, some systems make no serious attempt at "security by obscurity", instead they essentially depend on the honor system. Such a system can only enforce security on the most naive users, while giving a false sense of security to those who rely on the system to protect valuable content.

Perhaps due to their reliance on weak or nonexistent security measures, DRM companies are reluctant to make details of their systems public. This is understandable since many systems would likely be exposed as borderline fraudulent while even those that make a serious effort at security might risk weakening the system's already limited security.

In the field of security, experience has taught that full disclosure is essential. For example, cryptographers do not trust a cryptosystem until it has been publicly vetted and subject to intense scrutiny by the cryptographic community. This reluctance to accept cryptographic algorithms at face value comes from the long list of "secure" algorithms that have been shown to be insecure—knapsack cryptosystems are one interesting example (Odlyzko, 1990). In DRM there is, as yet, no such imperative to make the workings of systems, even in a general form, available for scrutiny. At the very least, this tends to suggest that the level of security actually provided by current DRM systems is suspect, since those making the security claims have a financial interest in boosting their perceived level of security.

The next section provides a reasonably detailed account of a carefully designed and implemented DRM system. This system has measures that attempt to prevent attacks at all levels. The system was built from the ground up, with security in mind, and it was careful reviewed at each stage of development. This may be the most detailed account of a functioning commercial DRM system yet to appear in public.

This paper then turns its attention to a few of the current crop of DRM systems. Unfortunately, it is extremely difficult to get any sense of the actual security measures employed by these systems. Again, this level of secrecy should raise the suspicions of potential users.

4. MediaSnap DRM

4.1 Overview

This section presents an overview of the MediaSnap DRM system (MediaSnap, 2003). This system was designed to protect PDF documents, though most of the same principles could be applied to audio, video or any other digital media. The system uses an email interface, with a pull-down menu to select the level of persistent protection. The email and attachments are encrypted using standard techniques, then sent to a server where the recipient's authentication information (password and/or certificate and/or biometric, etc.) is stored. The desired level of persistent protection is applied and the appropriate authentication information is included. The protected document is then sent to the recipient.

Once received, a protected document must be opened with an Adobe PDF reader and a MediaSnap PDF plugin. The heart of the DRM system lives in this PDF plugin.

Below, each of the primary security features of the MediaSnap DRM system is briefly discussed. At a high level, the DRM system can be viewed as consisting of two layers of protection, with various other security features built around these layers. At the outermost layer, the compiled code is encrypted and an anti-debugging technique is employed. While not invincible, the combination of these two features provides significant protection against all but dedicated attackers.

The second layer of protection includes a variety of security mechanisms, including standard encryption, proprietary key management techniques and non-standard scrambling of the data. These combine to make a challenging reverse-engineering problem once the outer layer of protection has been penetrated.

Other security features briefly discussed below include sophisticated tamper checking of software modules (to ensure that only the expected code is actually running), a low level anti-screen capture technique (to prevent the most obvious attack on digital documents), watermarking (to give some chance of post-theft prosecution) and a modest attempt to make each version of the software unique.

4.2 Tethered versus Untethered

Before discussing details of the MediaSnap security features, one crucial consideration in the design of a DRM system needs mentioning: whether the system will be tethered or untethered. In a tethered system, the cryptographic keys are kept on a server. Whenever the document is to be accessed, the key is transmitted from the server to the client machine. The key is then used to access the document, and the key is then discarded. This implies that an active network connection is required whenever the document is to be accessed. In the untethered case, the key is kept with the document at all times. Consequently, the document is accessible to the user even when no network connection is available.

The tethered case is generally considered slightly more secure since the keys are only briefly exposed when the document is opened, while in the untethered case the keys are always with the document and hence vulnerable to an attack "offline." The tethered case offers the benefit that keys can be managed on the server. In this case, if a key is deleted from the server, the document is effectively "shredded"—assuming that the document was not previously compromised. Of course, the untethered case is more convenient since the user can access the document at any time, without first making a connection to a key server. And it must be emphasized that even in the tethered case, the key is in the hands of the user (and potential attacker) at some point in time whenever the document is accessed.

Most systems only operate in a tethered mode, Authentica being perhaps the best-known example. The MediaSnap system is able to operate in either a tethered or untethered mode. The discussion below is focused on the untethered mode, since it is the more general case.

4.3 Encrypted Object Code

In the MediaSnap system, the object code is encrypted in blocks, where each block could be as small as a "basic block" or as large as individual functions or even larger units. There is some tradeoff between block size (smaller being somewhat more secure, since one compromised block does not necessarily imply that all blocks are compromised) and execution speed (smaller block size will tend to slow execution). The code is then decrypted as it executes, with only a single block in the clear at any given time. Tamper checking is employed at this point, providing a measure of "fragilization" comparable to the methods discussed in Horne, Matheson, Sheehan, and Tarjan (2002), and Chang and Atallah (2002). This "mix-mastering" of the code is intended to serve a similar purpose as the method described by Aucsmith (1996).

4.4 Debug Disrupter

From the relatively protected space of the currently decrypted code block, an anti-debugger is called. This program is designed to detect programs that try to set breakpoints. If this debug disrupter finds that a breakpoint has been set, it will cause the DRM application to close.

Note that the decrypted segment of code is protected from inspection by the debug disrupter, while the decrypted code block calls the debug disrupter. These two interlocking mechanisms perhaps offer greater security in combination than the sum of the security provided by the two component parts.

4.5 Content Encryption

The MediaSnap system employs the Advanced Encryption Standard (AES) for content encryption. Any other well-known, secure crypto algorithm would, of course, suffice.

As discussed above, encryption is a necessary step, since it eliminates the possibility that an attacker can convert the protected document into plaintext without attacking the DRM software. However, standard implementations of crypto-algorithms are easily detected in software, making it relatively easy to extract the keys, while non-standard implementations of such algorithms are not as trustworthy, and probably not as efficient. Furthermore, if an attacker does succeed in capturing the cryptographic key, he can then use any implementation of

the well-known crypto-algorithm to decrypt the content, thus avoiding the potentially difficult task of reverse engineering this crucial piece of the security software.

4.6 Scrambling

In order to tie all of the content to an unknown algorithm (from an attacker's perspective), the MediaSnap system employs a proprietary "scrambling" algorithm. This algorithm is actually a large family of related crypto-algorithms. The algorithm has been tested extensively and no obvious weakness has been found. Of course, the algorithm has not undergone the rigorous peer review necessary before it could be considered secure, but the purpose of the algorithm is not to provide additional cryptographic strength. Instead, the purpose is to force an attacker to do more than simply recover the cryptographic key. Therefore, as long as the algorithm provides a modest amount of cryptographic strength, it serves its purpose.

The plaintext is scrambled then encrypted, as illustrated in Figure 2. In this way, no properties of the AES algorithm are affected, since we have, in effect, simply changed the plaintext. Also, when the protected data are decrypted, they are still in a scrambled form. Consequently, an attacker who recovers the cryptographic key and decrypts the data will obtain scrambled data, not plaintext. Hence, the attacker must defeat the scrambling in order to recover any plaintext. Since the scrambling algorithm is proprietary, recovering the scrambling keys without recovering the algorithm is insufficient to recover the plaintext. And since there exists a large family of scrambling algorithms, this is a place where uniqueness (discussed below) can easily be applied. Finally, there is additional freedom in manipulating the scrambling keys, making it potentially more difficult for an attacker to recover these keys as compared to the cryptographic keys.

Figure 2: Scrambling and encryption

4.7 Authentication

The MediaSnap authentication mechanism allows for arbitrary Boolean combinations of passwords, biometrics, digital certificates, etc. For example, a document could be protected so that it would open with either user A's password or user B's thumbprint and certificate. The MediaSnap approach to authentication does not return a 1-bit (yes/no) result. Instead it returns a value—regardless of whether the authentication was valid or not—which is used in a subsequent computation. If an invalid authentication value is entered, this computation will yield an invalid result and the document will not open. When such a failure to open occurs, it is far removed from the point in the software where the invalid data first appeared. In order to break this authentication mechanism, an attacker needs to do more reverse engineering than simply flipping a single bit.

In general, it is perhaps bad practice to ever have a single bit that can be flipped to subvert a crucial security mechanism. As an aside, this seems to be an inherent weakness of current off-the-shelf biometric authentication methods, such as thumbprints.

4.8 Key Management/Caching

The key management problem is critical to any DRM system. Somehow, a secret must be shared between the creator of the protected content (the server) and the software that will render that content. Public key cryptography is one way—though certainly not the only way—to share such a secret.

In the MediaSnap system, the cryptographic key, Kc, must be passed to the player. The key Kc is encrypted with another key, Km, where Km depends on three key parts, one from the player, one from the content and one from the user.

Protecting a secret stored in software is the fundamental challenge of DRM. Suppose that S is a secret a server wants to protect. Since the software consists only of code and data, S = F(D) for some code F operating on some data D, where D can be empty. Obviously, if an attacker knows both F and D, then S is known. Consequently, one desirable property is that S is difficult to recover when either F or D is known, but not both. Another consideration is to make it non-obvious that the data D are hiding a key. Shamir and van Someren (1998) show that keys themselves are easy to detect simply because they are random data, while most software and data are highly structured. This leads to the somewhat counterintuitive result that D should not appear to be "too random".

Caching is a closely related problem. For example, a server might not want to ask the user to re-authenticate each time the authentication result is required. Therefore, it would be useful to cache the authentication value so that

it can be retrieved whenever it is required. The MediaSnap software employs a scheme where each time data are cached, a different data value is stored into a different location in memory.

4.9 Module Tamper Checking

The MediaSnap system supports tamper checking of all software modules that are loaded when the application is running. If an unapproved module is detected, the document cannot be viewed. Note that a sophisticated use of such tamper checking can frustrate certain standard attacks, such as running the software under VMWare. In practice, an extremely high level of tamper checking can be difficult to achieve since different systems will tend to have vastly different software running. Hence, a more limited degree of tamper checking—involving only the security-critical modules—is also supported.

4.10 Anti-Screen Capture

Perhaps the most obvious way to attack a DRM document protection system is to use a screen capture program to capture the image and save it to an unprotected file. The MediaSnap system employs a low-level utility that intercepts calls related to screen capture. In this way, it is possible to filter legitimate screen capture operations from those that attempt to access screen memory currently in use by the protected application.

One attack against such an anti-screen capture utility is to intercept the information destined for the anti-screen capture utility and always send information indicating that no screen capture is occurring. This is, essentially, a classic "man-in-the-middle" attack, which is also an attack that could be used on the debug disrupter. Consequently, the design of the anti-screen capture mechanism can be similar to that of the debug disrupter.

Of course, an attacker can always use a digital camera to capture the image (the so-called "analog hole," an inherent weakness in any DRM system; EFF Consensus at Lawyerpoint, 2002). If this occurs—or the document protection is broken by any other means—the last line of defense lies in the realm of digital watermarking.

4.11 Digital Watermarking

An invisible digital watermark potentially enables us to track the source of a stolen document. For this reason, each MediaSnap protected document includes such a watermark. Unfortunately, the current state of watermarking technology is such that if an attacker knows the watermarking technique, he can almost certainly remove the mark, or at least mangle it beyond recognition. It is therefore necessary that the MediaSnap watermarking scheme remain secret—see Craver et al. (2001), for an account of the perils of using a known (or easily discovered) watermarking scheme.

4.12 Uniqueness

Several of the security features described in the previous sections can be made unique to each instance of the DRM player software. There are two different levels of uniqueness that are of interest. At one level, each player could have functionally unique features in which case a piece of content must be processed differently for each player. For example, if a different scrambling algorithm is included in each player then a document protected for one particular player will not be readable by any other player. Of course, this requires that the server create a protected document specifically for a particular player. This functional level of uniqueness provides a high degree of security, but it also creates some problems. For example, forwarding documents is difficult.

By varying only those functions internal to the DRM software a lesser degree of uniqueness can be achieved. For example, the internal key management within each player could be unique. This level of uniqueness does not affect the functionality in any way, but it is nevertheless likely to create a more difficult problem for an attacker.

The potential benefits of uniqueness or "genetic diversity" in software have previously been noted (Cohen, 1992; Forrest, Somajaji, and Ackley, 1997). In the context of DRM, uniqueness creates a system where an attacker may be able to break one particular instance of the software without necessarily breaking the entire system. Strangely, within the DRM community software uniqueness appears to have had little impact outside of MediaSnap and Macrovision (Macrovision, 2000). The (non-DRM) company most aggressively pursuing software uniqueness appears to be Cloakware (Cloakware, 2002; Nickerson, Chow, Johnson, and Gu, 2001).

For the MediaSnap DRM system, each player includes both a generic (or universal) copy of these functions as well as a unique set. Consequently, a document can be protected so that it is readable using any copy of the MediaSnap software or so that it is only readable by one particular instance of the software. This offers the benefits of both a universal reader and a unique reader in a single package.

4.13 Usage Data Dilemma

In an untethered mode, keeping track of usage data (e.g., the number of times the document has been read), or logging information, presents a major challenge. The usage data could, for example, be tied to the user, the hardware or the document. If they are tied to the user, then if user A has a document that can only be opened, say, three times, user B might also be able to open it three times, provided user A gives user B the required authentication information. This "other" user could, of course, also be user A or user A's spouse, etc. If instead, the usage data are tied to the document, then user A might be able to do a replay attack by simply restoring the document that was

downloaded, in effect resetting the usage data to their initial state. If the usage data are tied to the hardware, this might prove undesirable to users who want to be able to read the document on any of several different platforms (PC, Palm, etc.).

In tethered mode, such data can be kept on the server. This is perhaps the single greatest advantage of tethering.

4.14 Other Security Features

There are at least two additional security approaches worth mentioning. First is the idea of software obfuscation (Collberg, Thomborson, and Low, 2000). Heavy use of obfuscation would likely increase the time required to reverse engineer the software. However, Barak et al. (2001) make it clear that obfuscation is far from a panacea.

A second security approach is software "fragilization", meaning that small perturbation in the software tend to cause the software to fail (Horne et al., 2002; Chang and Atallah, 2002). (As an aside, it is interesting to note the many similarities between Horne et al., 2002, and Chang and Atallah, 2002, even thought they are both clearly patented.) This is an interesting idea since it is likely to increase the work required by an attacker. The MediaSnap system employs this concept to some extent in combination with encrypting the object code. In addition, fragilization could be applied to the MediaSnap software directly, so that even if the code encryption is defeated, the software remains fragile.

5. Other DRM Systems

This section briefly discusses several other implemented DRM systems. The information comes from press releases, white papers and rudimentary observations of the systems.

5.1 Exaggeration

Here is a selection from the 2002 version of Atabok's Web site (Atabok, 2002):

> Q. How does Atabok's security compare to the competition?
> A. The majority of service providers offer the ability to encrypt at 128-bits. Atabok encrypts your content with 256-bit encryption, which is exponentially more secure. Additionally, Atabok has the approval from the Department of Commerce to export at this level, something others cannot do.

A similar example (from Authentica) is given by Schneier (2000). In general, DRM companies appear to be obsessed with encryption and, in particular, key length. But potential consumers of these products should be very wary of companies that tout encryption to the exclusion of other necessary DRM features. As we have seen, encryption is the easy part of any comprehensive DRM method. Or, as Schneier (2000) succinctly puts it:

> What does breaking the encryption have to do with breaking the system? Haven't these people learned anything from the DeCSS story?

5.2 Pure Hype

A secretSeal press release (secretSeal, 2000) claims their product "contains five radical innovations", which are "(1) hieroglyphic passwords (ancient Egyptian hieroglyphics!), (2) variable-length encrypted keys, (3) a morphogenetic encryption algorithm, (4) no encryption formula present in the software and (5) the use of public keys".

Apparently, even secretSeal is embarrassed by this press release, since they have no link to it from their current website. Unfortunately, their current website also gives no hint as to what they are now doing.

5.3 Respect

Companies have fielded DRM systems whose security is designed to defeat a totally naive user, while making no attempt to prevent a moderately skilled attacker from breaking the system. Adobe eBooks is an example of such a system. Adobe documentation (Adobe, 2003) states (p. 67): "It is up to the implementors of PDF viewer applications to respect the intent of the document creator by restricting access to an encrypted PDF file according to passwords and permissions contained in the file." Guignard (2003) says succinctly: "Because of the 'respect' model used by the PDF specification, your encrypted PDF files offer about as much security as dried eggshells!" Of course, breaking these "dried eggshells" led to the Digital Millennium Copyright Act (DMCA; DMCA, 2000) prosecution against Dmitry Sklyarov and ElcomSoft (EFF DMCA, 2002).

Bailey (2001), writing about the security of eBooks, makes the valid point that "PCs inherently offer no way to protect secrets." Perhaps this was Adobe's rationale for adopting their respect model—essentially the honor system for programmers.

It is interesting to compare Adobe's approach to the MediaSnap system. In the latter, a serious effort was made to limit a legitimate user's ability to attack critical parts of the system. Of course, skilled attackers who devote sufficient effort will almost certainly break the security on a specific piece of content. However, the uniqueness built

into the MediaSnap system offers some hope that the system can survive, even after repeated successful attacks. On the other hand, the approach followed by Adobe insures that serious attackers will succeed quickly and that any break will almost certainly destroy the security of the entire system.

5.4 Patently Obvious

InterTrust is an interesting DRM company. They were perhaps the earliest entrant into the DRM marketplace. At the time of this writing, the company holds 24 patents along with another 80 patent applications (InterTrust, 2002). InterTrust is currently suing Microsoft for patent infringement (Patently Obvious, 2001).

One patent that is at the heart of InterTrust's lawsuit against Microsoft is United States patent 6,185,683. This monstrous patent runs to 250 printed pages, with more than 200 (mostly cartoonish) diagrams. The diagram reproduced in Figure 3 appears on the first page of the patent and nicely summarizes its content.

A careful reading of this patent is, perhaps, instructive of InterTrust's strategy in the DRM marketplace. Apparently, the diagram in Figure 3 is meant to illustrate the electronic version of a traditional "secure courier", where a trusted human courier delivers a message. The entire patent (along with its sister patent, 6,253,193) make no serious attempt to deal with any of the technical issues involved in a DRM system. Instead, the goal appears to be to lay claim to the digital version of the traditional secure courier.

In spite of all of their patents, it is not clear what technology, if any, InterTrust actually employs in its DRM products. The only specific technical details deal with cryptography, which, as discussed above, is only a small part of any reasonably complete DRM approach. Even the self-serving white paper (Duhl and Kevorkian, 2002) makes the point that "technical credibility" is a crucial factor when selecting a DRM product. Unfortunately, the paper then leaps to the illogical conclusion that InterTrust's patent portfolio is a sure sign of their technical credibility. Business process patents, such as number 6,185,683, contribute nothing towards demonstrating the technical prowess (or lack thereof) of InterTrust's DRM system (Boulton, 2001).

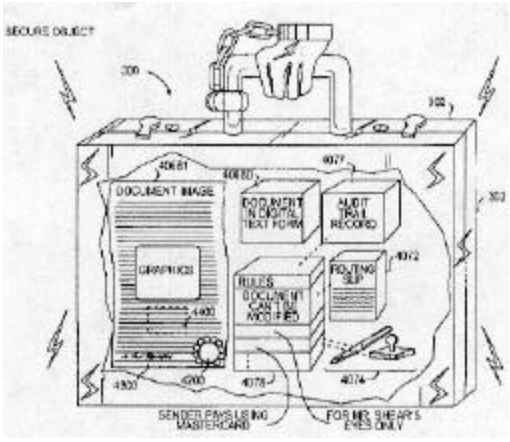

Figure 3: InterTrust patent

5.5 Kernel Mode

A plausible DRM approach is for the software to "take control" at the OS kernel level. The anti-debugging and anti-screen capture methods described in the MediaSnap section, above, operate at such a low level.

The MediaRights system from Elisar Software claims to provide security at the OS kernel level. However, a careful reading of an overview of the system (Elisar, 2001) indicates that the authors seem to believe that by decrypting in kernel mode instead of user mode, they have achieved meaningful DRM protection. But unless a serious attempt is made to harden the entire system against attack, this relatively minor modification is likely to be of little practical significance.

6 Conclusion

Bettors at the horse races like to say: "You can beat the race, but you can't beat the races." With any combination of current DRM technology, it is possible for a skilled attacker to "beat the race", that is, to defeat the persistent protection applied to a given piece of digital content. Of course, certain combinations of the techniques will provide more of a challenge than others. However, at the current evolutionary stage of DRM there appears to be no escape from the dubious concept of "security by obscurity." Consequently, once a dedicated attacker has gone through the (perhaps tedious) process of removing the obscurity, the security vanishes as well. Furthermore, it appears that for nearly all systems currently on the market, "beating the race" is equivalent to "beating the races". In

other words, a successful attack on one particular instance of the DRM software immediately applies to all instances of the software; see Guignard (2003) and Beale Screamer (2001) for examples of such DRM systems.

Tamper-resistant hardware is the most obvious solution to the DRM problem. The Trusted Computing Group, or TCG (TCG, 2003), and the Next-Generation Secure Computing Base, or NGSCB (NGSCB FAQ, 2003), are industry attempts to include such hardware in a future generation of personal computers, primarily for DRM purposes. The TCG/NBSCB initiative—formerly known as TCPA/Palladium (EPIC, 2002)—has the backing of Intel, Microsoft and many other leading technology companies. Though such a hardware-based approach clearly offers many security advantages, it also raises many troubling issues (Anderson, 2003). For example, it threatens to move the desired "remote controllability" from users to third parties.

If DRM is to be successful in software-based systems, perhaps the best hope lies in the realm of software uniqueness. If each instance of a particular DRM software product includes some degree of uniqueness then an attack that succeeds against one will not necessarily succeed against all. Thus, "beating the race" would not automatically "beat the races." In fact, the problem of defeating all instances of unique software is potentially many orders of magnitude more difficult than defeating a single instance. Of course, much depends on how and where the uniqueness is employed.

It is conceivable that the idea of uniqueness could take software-based DRM beyond its current level of hype—which is bound to disappoint as fielded systems continue to fail—to a level where a slow leak of protected content is an accepted fact, but complete and total failure of systems is a rarity.

ACKNOWLEDGMENT

I want to thank Joe Pasqua, Paul Clarke, Jason Rhoads, Louis DeMartini and all at MediaSnap, Inc., who designed and developed a first-rate DRM system—and allowed me to write about it. And I would like to thank an anonymous referee whose insightful comments greatly improved several sections of this paper.

REFERENCES

Adobe, PDF Reference, second edition, Adobe Portable Document Format Version 1.3, 2003, http://partners.adobe.com/asn/acrobat/docs.jsp, Last accessed July 28, 2003 (registration required).

Anderson, R.J., *Security Engineering: A Guide to Building Dependable Distributed Systems*, John Wiley & Sons, Inc., Chapter 23, 2001.

Anderson, R.J., "Trusted computing frequently asked questions – TCPA/Palladium/NGSCB/TCG," 2003, http://www.cl.cam.ac.uk/%7Erja14/tcpa-faq.html, Last accessed July 28, 2003

Atabok, "Answers To Frequently Asked Questions," 2002.

Aucsmith, D., "Tamper resistant software: An implementation," *Proceedings of the First International Information Hiding Workshop, Lecture Notes in Computer Science 1174*, Springer-Verlag, Heidelberg, Germany, pp. 317-334, 1996.

Bailey, D.V., Inside eBook security, *Dr. Dobb's Journal*, November 2001, http://www.ddj.com/documents/s=1487/ddj0111d/0111d.htm, Last accessed July 28, 2003 (registration required).

Barak, B., O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. Vadhan, and K. Yang, "On the (im)possibility of obfuscating programs (extended abstract)," *Advances in Cryptology - CRYPTO '01, Lecture Notes in Computer Science 2139*, J. Kilian (ed.), Springer-Verlag, Heidelberg, Germany, pp. 1-18, 2001, http://www.eecs.harvard.edu/~salil/papers/obfuscate-abs.html, Last accessed July 28, 2003.

Beale Screamer, Microsoft's digital rights management scheme — technical details, 2001, http://cryptome.org/ms-drm.htm, Last accessed July 28, 2003.

Boulton, C., "InterTrust Claims .NET, XP Infringe on Patents," *internetnews.com*, 2001, http://www.internetnews.com/bus-news/article.php/3_906191, Last accessed July 28, 2003.

Casti, J. L., *The One True Platonic Heaven: A Scientific Fiction of the Limits of Knowledge*, Joseph Henry Press, Washington, DC, 2003. Available in electronic form from National Academies Press at http://www.nap.edu/catalog/10533.html, Last accessed July 28, 2003.

Chang H. and M. J. Atallah, "Protecting software code by guards," *Security and Privacy in Digital Rights Management, ACM CCS-8 Workshop DRM 2001, Philadelphia, PA, USA, November 5, 2001, Revised Papers. Lecture Notes in Computer Science 2320*, T. Sander (Ed.), Springer-Verlag, Heidelberg, Germany, pp. 160-175, 2002.

Cloakware, "'Genetic diversity' as a defense against automated attacks on software," 2002, http://www.cloakware.com/pdfs/Genetic Diversity 06-02.pdf, Last accessed July 28, 2003.

Cohen, F.B., "Operating system protection through program evolution," 1992,

http://all.net/books/IP/evolve.html, Last accessed July 28, 2003.
Collberg, C., C. Thomborson and D. Low, "A Taxonomy of Obfuscating Transformations," Technical Report #148, Department of Computer Science, The University of Auckland, Auckland, New Zealand, 2000, http://www.cs.arizona.edu/~collberg/Research/Publications/CollbergThomborsonLow97a/index.html, Last accessed July 28, 2003.
Craver, S.A., M. Wu, B. Liu, A. Stubblefield, B. Swartzlander, D.W. Wallach, D. Dean, and E.W. Felten, "Reading between the lines: Lessons from the SDMI challenge," *Proceedings of 10th USENIX Security Symposium*, August, 2001, available from http://www.cs.princeton.edu/~felten/, Last accessed July 28, 2003.
Cryptographers Panel, *RSA Conference 2002*, 2002, http://www.rsaconference.com/rsa2002/, Last accessed July 28, 2003
DMCA, "Digital millennium copyright (DMCA) information," 2000, http://www.tuxers.net/dmca/, Last accessed July 28, 2003.
Duhl J., and S. Kevorkian, "Understanding DRM systems: An IDC White Paper," 2002, http://www.intertrust.com/main/research/index.html, Last accessed July 28, 2003 (registration required).
Earley, P., "Family of spies: The John Walker Jr. spy case," *The Crime Library*, 2003, http://www.crimelibrary.com/terrorists_spies/spies/walker/1.html, Last accessed July 28, 2003.
EFF Consensus at Lawyerpoint, "Hollywood wants to plug the 'analog hole'," May 23, 2002, http://bpdg.blogs.eff.org/archives/000113.html, Last accessed July 28, 2003.
EFF DMCA, "US v. ElcomSoft & Sklyarov FAQ," February 19, 2002, http://www.eff.org/IP/DMCA/US_v_Elcomsoft/us_v_elcomsoft_faq.html, Last accessed July 28, 2003.
Elisar, "An overview of digital rights enforcement and MediaRights™ technology," Elisar Software Corporation, April 5, 2001, http://www.elisar.com/news/MRoverview.pdf, Last accessed July 28, 2003.
EPIC, "Microsoft Palladium," Electronic Privacy Information Center, November 11, 2002, http://www.epic.org/privacy/consumer/microsoft/palladium.html, Last accessed July 28, 2003.
Forrest, S., A. Somayaji, and D.H. Ackley, "Building diverse computer systems," *Proceedings of The 6th Workshop on Hot Topics in Operating Systems*, IEEE Computer Society Press, Los Alamitos, CA, pp. 67-72, 1997, available at http://www.cs.unm.edu/~forrest/publications/hotos-97.pdf, Last accessed July 28, 2003.
Geer, D., "Who will kill online privacy first—The lawyers or the techies?," *RSA Conference 2002*, http://www.rsaconference.com/rsa2002/, Last accessed July 28, 2003.
Guignard, B., "How secure is PDF?," 2003, http://www.cs.cmu.edu/~dst/Adobe/Gallery/PDFsecurity.pdf, Last accessed July 28, 2003.
HIPPA HQ, "HIPPA Resources," 2003, http://www.advanceforhim.com/common/Editorial/editorial.aspx?CC=659, Last accessed July 28, 2003.
Horne, B., L. Matheson, C. Sheehan, and R.E. Tarjan, "Dynamic self-checking techniques for improved tamper resistance," *Security and Privacy in Digital Rights Management, ACM CCS-8 Workshop DRM 2001, Philadelphia, PA, USA, November 5, 2001, Revised Papers, Lecture Notes in Computer Science 2320*, T. Sander (Ed.), Springer-Verlag, Heidelberg, Germany, pp. 141-159, 2002
InterTrust, "InterTrust intellectual property," 2002, http://www.intertrust.com/main/ip/index.html, Last accessed July 28, 2003.
Macrovision, "Software tamper-proofing deployed 2 year anniversary report," September, 2000, http://www.cptwg.org/Assets/September%20presentations/Software_Tamper-Proofing.ppt, Last accessed July 28, 2003.
MediaSnap, "Welcome to MediaSnap," MediaSnap Inc., San Jose, CA, 2003, http://www.mediasnap.com, Last accessed July 28, 2003.
NGSCB FAQ, "Microsoft Next-Generation Secure Computing Base — Technical FAQ," February, 2003, http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/news/NGSCB.asp, Last accessed July 28, 2003.
Nickerson, J.R., S.T. Chow, H.J. Johnson and Y. Gu, "The encoder solution to implementing tamper resistant software," *CERT/IEEE Information Survivability Workshop*, Vancouver, Canada, October 2001. http://www.cert.org/research/isw/isw2001/papers/Nickerson-12-09.pdf, Last accessed July 28, 2003.
Odlyzko, A.M., "The rise and fall of knapsack cryptosystems," *Cryptology and Computational Number Theory: Proceedings of the 42nd Symposium in Applied Mathematics*, C. Pomerance (ed.), American Mathematical Society, pp. 75-88, 1990, http://www.dtc.umn.edu/~odlyzko/doc/arch/knapsack.survey.pdf, Last accessed July 28, 2003.
Patently Obvious, "Intellectual Property Analysis of InterTrust Technologies' U.S. Patent No. 6,185,683 InterTrust Technologies Corp. v. Microsoft Corp.," May 11, 2001,

http://www.m-cam.com/patentlyobvious/20010511_intertrust.pdf, Last accessed July 28, 2003.
PDFzone.com, "Digital rights management: A primer," December 12, 2001, http://www.pdfzone.com/news/100864.html, Last accessed July 28, 2003.
Rosenblatt, B., B. Trippe and S. Mooney, *Digital Rights Management: Business and Technology*, John Wiley & Sons, Inc., Indianapolis, IN, 2001.
Sayer, P., "Stephen King's 'Plant' withers," *The Industry Standard*, November 29, 2000, http://www.thestandard.com/article/display/0,1151,20433,00.html, Last accessed July 28, 2003.
Schneier, B., "The doghouse: Authentica," *Crypto-Gram*, August 15, 2000, http://www.counterpane.com/crypto-gram-0008.html, Last accessed July 28, 2003.
secretSeal, secretSeal Technologies Corporation press release, 2000, http://www.google.com.au/search?q=cache:M3WVGiKYes8J:www.arenatech.com/portfolio/pressrel.pdf, Last accessed July 28, 2003.
Shamir, A. and N. van Someren, "Playing hide and seek with stored keys", September 22, 1998, http://www.web.ms11.net/hawaii/keyhide2.pdf, Last accessed July 28, 2003.
TCG, Home Page of Trusted Computing Group, 2003, http://www.trustedcomputinggroup.org/home, Last accessed July 28, 2003.
von Lohmann, F., "Reconciling DRM and fair use: Preserving future fair uses?," *12th Conference on Computers, Freedom & Privacy*, San Francisco, CA, April 16-19, 2002, http://www.cfp2002.org/fairuse/lohmann.pdf, Last accessed July 28, 2003.