Digital Rights Management (DRM) and Secure Systems Expert Witness: Brief (but Broad) Highlights of the History of Digital Rights Management and the Like

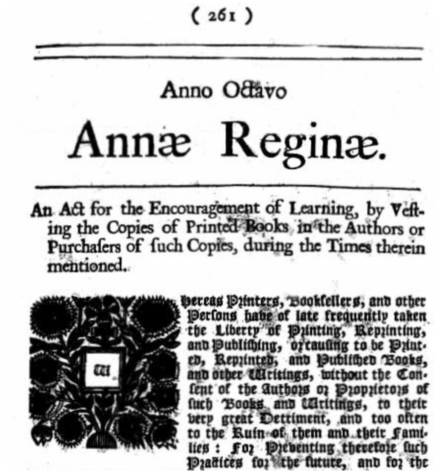
By Jean Renard Ward

Digital Rights Management ("DRM") and secure software systems may be at issue in litigation concerning patent rights. An expert witness on Digital Rights Management technologies, especially in light of the long history in this area, may be helpful in these matters.

More than one person has commented that DRM does not stand for "digital *rights* management", so much as for "digital *restrictions* management". The concern is really about controlling things you *don't* want people to do. The distinction is subtle, but important. As a business matter, you may need to trade off

- how much a failure alienates honest customers (when it fails, it says "**NO**")
- allowing undesired rights to the not-so-honest (when it fails, it says "**Yes**").[i]

So as not to be too confusing, we will just say "rights management".



(261)

Anno Octavo

Annæ Reginæ.

An Act for the Encouragement of Learning, by Veſting the Copies of Printed Books in the Authors or Purchaſers of ſuch Copies, during the Times therein mentioned.

Whereas Printers, Bookſellers, and other Perſons have of late frequently taken the Liberty of Printing, Reprinting, and Publiſhing, or cauſing to be Printed, Reprinted, and Publiſhed Books, and other Writings, without the Conſent of the Authors or Proprietors of ſuch Books and Writings, to their very great Detriment, and too often to the Ruin of them and their Families: For Preventing therefore ſuch Practices for the future, and for the

British Copyright Act of 1710[ii]

The history of rights management or control is ancient. It is the philosophical basis for public copyright law, a legal innovation that dates to the 17th/18th centuries. This true innovation (it had never existed before, and someone had to think it up) was in response to the wide use of the printing press to Europe.[iii] The printing press made the physical copying of written works much less expensive, so it was easier to “steal” a book by making a copy. This is very analogous to the modern digital age. Physical copying of things in electronic form essentially costs nothing. Cheap copying has been the continual driver for innovations in DRM technologies.

Rights management in all forms generally involves a

- *restriction* mechanism (think: a lock) associated with or attached to the thing you want to protect; a related
- *enablement* mechanism (think: a key) associated with the authorized user; and a
- *management* mechanism (think: some rules), to control which things the user can do. (Sometimes this is built into the enablement mechanism.)

Implicit in this, of course, is that the user is *authentic*, that you are not dealing with an impostor. Users can be authenticated with something they know – such as a password – something they have – such as an ID card, a security token, or a cell phone with a particular number – or by some property of the user – often a biometric property, such as the look of their face, the sound of their voice, their signature, or their fingerprint.

For legal copyright,

- the restriction mechanism can be something as simple as a one-line copyright notice attached by printing it in a book, or associated by registering a book at a national library,
- the enablement (disablement?) mechanism is the threat of suing the user, and
- the management mechanism is a signed contract or license, with specific terms.

But rights management is much older even than legal copyright. Simply locking something in a box is a form of rights management.

- The restriction mechanism is the lock on the box: it is associated with the object when the object is put into the box. Sometimes the lock was ***literally*** attached to the book, like a teenager's diary.
- The enablement mechanism is the key the person has or is allowed to use.'
- The management mechanism was controlling who got the key.

For another medieval example,

- the restriction mechanism might be a chain that was literally attached to the book.[iv]
- the enablement mechanism was letting someone into the room (or not),
- the management mechanism was controlling who you let into the room.

Digital rights management techniques still generally follow the same model.

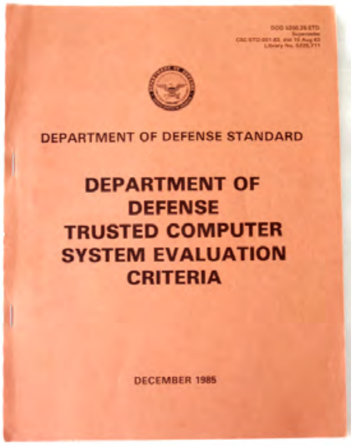
- A restriction mechanism is associated with the file or somehow "attached".
  For example, the file is stored in a secure database with an associated ID code, or the file is encrypted with an associated key or keys, or data is built into the file.
- An enablement mechanism is associated with or held by the user.
  For example, a password, a smartcard token, or a hidden license file on their device, with enforcement by a trusted and secure system.
- The management mechanism is trusted software using encoded control information.
  The control information might be attached somehow to the data file (along with depending on a trusted, secure software system to enforce the control), or checking back to a central server for permissions (again, enforced by a trusted, secure system).

Implementations for digital rights management have a long history. DRM systems using cryptographic techniques and centralized control were well known in the 1970's and 1980's.[v]

The reasons for looking to encryption are self-evident: even if someone could copy a file, they couldn't read it.

There were techniques for attaching/associating restriction mechanisms and management mechanisms directly to the file itself[vi], techniques for managing permissions over a network[vii], techniques for expressing legal contracts in a “rights definition language” and enforcing the terms with secure software[viii], techniques for temporary and/or portable licenses for roaming (and even traveling) users.[ix] There were multiple techniques for managing not just the rights of one user, but managing what rights could be passed from one user to another.[x]

Digital rights mechanisms depend on having computer systems that are *secure*: having a lock is somewhat useless if the lock is super-easy to pick. (Except perhaps for giving you legal basis to sue, because after all, somebody picked your lock: the DMCA for example made it illegal to bypass a copy-protection mechanism, no matter how wimpy.)[xi] Tamper-proofing a computer system (or making it tamper-evident: not quite the same thing) is essentially digital rights management for who can change the system. Encrypting the software (more than just "digitally signing": that is only the authentication part) is a powerful tool for this.[xii]


DEPARTMENT OF DEFENSE STANDARD
DEPARTMENT OF DEFENSE TRUSTED COMPUTER SYSTEM EVALUATION CRITERIA
DECEMBER 1985

Picture credit: AlephGamma[xiii]

A significant reference point in the history of secure systems and DRM was – and still is – the "Orange Book" (named from the color of the cover) from 1985.[xiv] This was essentially a very detailed "design guide" to building secure computer systems, and to managing rights to access computer files and data. This public document was widely distributed. Although funded by the US Department of Defense (as was much of the early public work on what became the internet), it was aimed specifically at spreading the techniques for rights management and secure systems broadly through commercial systems. It is still cited as a reference today.[xv]

The authors intentionally wrote the Orange Book as a *requirements* document: a detailed guide, rather than a follow-the-exact-steps "cook-book". As comprehensive as it was, it was carefully neutral, or agnostic, about the exact technologies that could be used. This was just good engineering practice: *not* being neutral could have the bad side effect of excluding other (and perhaps better) ways to do the same thing. (The verification procedures it spelled out for government procurement were, perhaps unfortunately, less “agnostic”)

The Orange Book also dealt in detail with controlling what rights or access one user could (and could not) pass on to another user. It described properties of a rights definition language (essentially, an electronic equivalent of the "contract" in copyright). There was pretty much no protection or rights-management need that fell outside the book: It went so far as to talk specifically about making mandatory the attachment of a restriction mechanism (e.g. like that copyright notice above) even to a simple print-out of a file.

The Orange Book was in turn based on concepts developed and implemented in the MULTICS project and operating system from the 1960s and 1970s. MULTICS featured prominently in my undergraduate engineering classes at MIT, since it was being developed on our campus.[xvi] MULTICS was a broad project with the explicit goals of actually building a *usable* secure general-purpose and multi-user system, that strongly protected what users could and could not do with data of all sorts. One of the engineering techniques to ensure that it ended up *actually* being usable was simply that we actually had to *use* it across the whole campus.

Authentication was part of MULTICS, and developers of MULTICS also concerned themselves with *biometric* authentication,[xvii] which only later "caught on" in many areas (think: fingerprint scanners on your laptop or tablet). Perhaps the earliest biometric technology known – long before the days of computers – was having someone sign their name, and a human-based recognition system (i.e. "eyeballs") verifying whether it looked like a known signature. Computer techniques for authenticating human-writing signatures dynamically ("on-the-fly") were also already well-known in the 1960s.[xviii]

Back to encryption: PKI.

Asymmetric encryption (commonly called PKI – Public/private Key Infrastructure) —with its one-way features, separate public and private keys, and "unforgable" digital signatures— became publicly known in 1976[xix] – though it was developed as a military secret earlier. Shortly after that, digital certificates were invented by a college undergraduate.[xx] PKI was adopted quickly into the existing repertoire of cryptographic DRM technology[xxi].

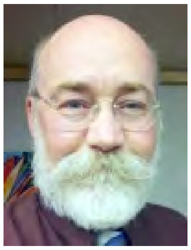
Speaking of "fingerprint":

In addition to biometrics, this term is also used for a "*digital fingerprint*". This refers to identifying a computer from a combination of incidental information: e.g. what combination of hardware happens to be on the computer, or what fonts and plug-ins happen to be installed in a browser. One use is to control files that a user simply copied to a different computer. Digital fingerprints have been used in DRM for consumer systems at least since the late 1990's to check whether a file or software has been copied to another computer, and also to identify who a user is (perhaps without the user knowing about it). Like many things related to DRM, there can be a trade-off between alienating customers (who perhaps happened just to have updated a disk drive) versus protecting against unauthorized copying.[xxii]

So, with all this history, all this prior art, just what is really new about DRM systems today? Or any of the associated technologies, like biometrics, encryption, and secure systems?

Or perhaps more significantly, what (if anything) is left that could be patentable today?

As patent practitioners know, it really depends on just what the words in a patent claim say, and how those words relate to *all* the prior art. A qualified expert witness in Digital Rights Management and the associated technologies can help in resolving a patent's claims.



About the Author: *Jean Renard Ward is highly experienced, MIT-educated expert witness in patent litigation. Mr. Ward's areas of design and development expertise include multi-touch/touchscreen and tablet hardware, capacitive touch and proximity sensors, styli/electronic pens, haptics; gestures, user interfaces (UIs), touchscreen graphics, and accessibility user interfaces (blind/visually-impaired); digital rights management (DRM), digital encryption and authentication (PKI), and malware detection; programming/coding (C/C++/Java, other systems), source-code analysis and reverse-engineering, and firmware. Clients include Google, Samsung, Ericsson, Lenovo, Motorola, Nokia, and Lucent Technologies. Mr. Ward has been granted multiple US patents. He received his degree in Computer Science and Electrical Engineering Degree from M.I.T.* ***Mr. Ward can be contacted at Rueters-Ward Services; Website: www.ruetersward.com Phone: (617) 600-4095; Cell: (781) 267-0156; Email:*** *jrward@alum.mit.edu*

[i] You may have experienced this first-hand for software on your PC: See for example
Fully Licensed GmbH, “Inside Windows Product Activation”, July 2001.
PC Review Newsgroups, “Product Activation gone berserk!”, September 4, 2003.

[ii] See “The Statute of Queen Anne”, http://www.copyrighthistory.com/anne.html

[iii] See Wikipedia.org: https://en.wikipedia.org/wiki/Johannes_Gutenberg

[iv] See http://www.nationalchurchestrust.org/what-see-inside/chained-bible

[v] Feistel, “Cryptography and Computer Privacy”, Scientific American, May 1973.

[vi] For example, by the encrypting the file. See also:
Pfleeger, “Security in Computing”, 1st ed., Prentice Hall, 1989.
Perritt, “Knowbots, Permission Headers and Contract Law”, Conf. On Techn. Strategies for Protecting Intellectual Property in the Networked Multimedia Environment”, April 1993.
Hartrick, EP Patent App. 93105502.4, November 3, 1993.

[vii] Kahn and Cerf, “An Open Architecture for a Digital Library System and a Plan for its Development”, Corp. for National Research Initiatives, 1988.

[viii] Kahn and Ely, US patent 6,135,646, October 24, 2000 (priority date 1993)

[ix] Fischer, US Patent 5,412,717, May 2, 1995: Hartrick, EP Patent App. 93105502.4, November 3, 1993.

[x] Wyman, US Patent, 5,204,897, April 20, 1993.

[xi] The text of the law can be found at https://www.copyright.gov/title17/92chap12.html#1201

[xii] Tyger and Lee, “Dyad: A System for Using Physically Secure Coprocessors”, Coalition for Networked Information, 1993.

[xiii] Picture source: AlephGamma at en.wikipedia - Transferred from en.wikipedia by SreeBot, Public Domain, https://commons.wikimedia.org/w/index.php?curid=16426265

[xiv] “Department of Defense Trusted Computer system Evaluation Criteria”, DOD 5200.28-STD, December 26, 1985. See especially the preface: “… to provide guidance to manufacturers as to what to build into their new, widely-available trusted commercial products…”

[xv] US Case No 2:14CV0061, (Eastern District of Texas, Marshall Division), 2015.

[xvi] Organick, “The Multics System: An Examination of its Structure”, MIT Press, 1972.

[xvii] Saltzer, “The Protection of Information in Computer Systems”, Proc. IEEE, September 1975.

[xviii] E.g. Peters,, US Patent 3,113,461, December 10, 1963; Danna, US Patent 3,480,911, November 25 1969.

[xix] Diffie and Hellman, “New Direction in Cryptography”, IEEE Transactions on Information Theory, Vol IT-22 no 6, November 1976.
Public-key cryptography was invented earlier, in the UK, but the discovery was kept secret: See Ellis, “The Possibility of Secure Non-Secret Communication”, CESG Research Report 3006, December 1969 (now declassified)

[xx] Kohnfelder, “Towards a Practical Public-key Cryptosystem”, Undergraduate Thesis, MIT, May 1978.

[xxi] Stefik, “Trusted Systems”, Scientific American, May 1997.

[xxii] HelpWithWindows.com, “Microsoft Windows XP Product Activation”, June 27, 2001.