

Technological Standards, Digital Rights Management and Free Software

***Are they compatible, or would implementing
the trusted computing standard within
the Linux kernel restrict the
'freedom' of free software***

By Paul Mobbs

An Essay for the Course on

Open Source/Free Software Philosophy and Theory (IA0260)

University of Göteborg

December, 2004

**Technological Standards, Digital Rights Management and Free Software:
Are they compatible, or would implementing the trusted computing standard
within the Linux kernel restrict the 'freedom' of free software.**

By Paul Mobbs. An Essay for the course on *Open Source/Free Software Philosophy and Theory* (IA0260), University of Göteborg. December 2004.

Abstract

Pressure from the developers of digital content is forcing those producing computer software to introduce systems for managing intellectual property rights. Whilst seeming to raise issues relating to copyright theft or computer security, trusted computing systems are challenging the free software movement to identify the purposes for which free software is created. Not only are trusted computing and digital rights management systems allowed under free software licensing, but there are already systems in existence to implement digital rights management within the Linux kernel. These changes are forcing a resolution to the debate that has existed between the grassroots and the corporate wings of the open source movement since the late-1990s. In the short-term the solution to this debate is likely to be a fork in the development of free software systems, but, in the longer term, free software may have to take on a political role that represents and protects the civil rights of computer users.

Introduction – Free Software and Digital Rights Management

This paper examines the impact of technological standards, in particular the current development of *trusted computing platforms* (TCP) that implement systems of *digital rights management* (DRM), on the development and use of free software. To understand the potential impacts of DRM on the free software community it is necessary to understand the technical implications of the new standards that are being developed to implement DRM, the effect that they have on the use of computer systems, and whether free software in its current form is compatible with the implementation of these standards.

The use of open technological standards, both in the hardware that executes the computer program and the data storage and communications standards that programs use to exchange information, have created a high level of *interoperability* (the capacity to swap information) between computer systems and the programs that people run on them. However, according to companies such as Microsoft, this era of open computing is coming to an end [NYT, 2002]. The development of standards that limit the ability of computer users to infringe the intellectual property rights of software writers, and the developers of the digital content that these programs use, has the potential to stifle the use of free software. This is because DRM not only affects the interoperability of computer systems, but also the functions within a computer system that the program code is able to utilise. Therefore the methods by which DRM systems are implemented could seriously affect the *freedom* of free software users.

The extent to which DRM is implemented within free software will depend primarily upon the attitude of the developers of these systems. A divide exists between those who view software as an exercise in engineering, and those who see it as having a political and social relevance too. Consequently, how DRM will affect the individual user of free software will be related to their position within this community, and by inference their personal view on the aims and objectives of free software. For this reason the introduction of DRM may precipitate a resolution to the conflict, perhaps long overdue, between those who view free software as a political issue at the heart of the emerging *Information Age*, and those who view it as merely a useful model for software development.

Content versus Code

For those who wish to retain open computing standards the problem with digital rights management is that its development is being motivated not by the *software* industry, but by the *content* industry (organisations who develop and market multimedia entertainment, video, music, literary works, etc.). Therefore the objectives of DRM are primarily related to protecting content of computer systems rather than improving the operational quality of those computer systems. This creates a fundamental difference of opinion, quite separate from issue of the *freedom of code*, between each side in the DRM debate. For example, those in the free software movement who cherish the autonomy that free software creates (such as Richard Stallman [Stallman, 2002]), or who lobby for privacy and civil rights within technological systems (such as the Electronic Frontier Foundation [Gilmore, 2001]), argue that DRM will restrict the rights of people to control their computers and to manage their privacy and security. In response those who promote DRM (such as IBM [Stafford, 2002a]) argue that the purpose of *trusted computing* is to control the security of computer systems, not to restrict the operation of programs.

Digital data, be it computer data or the DNA code in our body, can be copied repeatedly without degrading the quality of the information encoded within the data. In contrast analogue data, such as audio data on magnetic tapes or pages fed through a photocopier, are degraded each time they are copied, and successive copying can render the information unusable. For this reason one of the greatest benefits of digital networks to the general public, but for the owners of intellectual property perhaps the greatest liability, is that data can be endlessly copied across the network, at low cost, without loss of quality.

The concept of intellectual property rights, as they apply to *intangible commodities* such as songs, books or pictures, is that their use should create the same kinds of financial benefits that are created by the trading of tangible commodities such as copper, coffee or cars. For this reason the methods of controlling intellectual property rights that have arisen during the Industrial Age – principally copyright and contracts, but more recently patents – have crossed over into the digital domain. However, the ease with which digital information can be copied has rendered intellectual property rights on public data networks, especially those that operate across legal jurisdictions such as the Internet, virtually unenforceable.

The problem of controlling the online distribution of content has limited the extent to which electronic commerce has been able to develop. This is because the potential return on the

capital invested in the development of information systems and digital content is diminished by unlawful copying. Often to a point where the level of return on investment is not sufficient for those involved to consider investing in the development of new systems or content. For this reason the content industry have, as they did with previous *disruptive technologies* such as the video recorders [Intertrust, 2004], sought to introduce measures into computer networks to eliminate the threat of unlicensed copying.

Digital rights management has been in use for some time through the licensing of computer software. However computer software represents an *active* digital system, which can verify whether the computer user has legitimate access and terminate itself if the required validation fails. In contrast the vast majority of digital content is *passive*, and controlling its use presents a wholly different set of problems. The majority of content does not contain code, and so the extent to which its use can be controlled or limited is dependent upon the programs used to access the content. In any case, the fact that both the content and the programs that access the content are open to study by the computer user, and that user can also create their own programs, mean that the access control or copy protection system can be understood and circumvented.

Digital Rights Management and Trusted Computing

The basic principle of digital rights management is that the security layer, between the user and the digital content, is not controlled exclusively by a program or the design of the storage media. Instead the computer's operating system regulates access by identifying the legal status of the digital content and only allowing access to those who have the lawful right. This means that because it is the operating system that controls access, not the program, access to the programs and raw data is also restricted. This makes circumventing the security far harder (although, not impossible). Computer operating systems which incorporate design features to implement an agreed standard of digital rights management are called *trusted computing platforms* (TCP).

TCP systems control access to information within the computer through the use of *data encryption*, and access to the encrypted data is controlled using *encryption keys*. These keys allow access to certain files, or certain parts of the computer's operating system, by programs able to access the required keys. The use of encryption prevents the programs running on the computer from peeking at each other's data, unless that information is formally exchanged through the functions of the operating system.

In order that any operating system or program can be accredited as a *TCP application*, and access the encryption keys, it must be audited, and on the successful completion of the auditing process the software will be awarded a digital signature. The purpose of this signature is not just to verify that the software is TCP compliant, and that the program doesn't contain bugs or security flaws that might allow the unauthorised use of data. The digital signature is based in part upon the numeric content of the program code. Consequently any modifications to the code of the program can be detected by the operating system and the modified program will be refused access to trusted data. Whilst seemingly neutral, the problem with this system is that the auditing process costs a large sum of money. Not high

enough to stifle the development of new applications by proprietary software developers, but certainly high enough to make TCP accreditation unrealistic for the developers of small free software projects.

The development of TCP will enhance the economic power of the large online content providers [Anderson, 2003]. Whilst the testing and auditing of TCP applications might represent a solution to some of the problems related to bad code, viruses and the unauthorised use of data, even the supporters of the new standard admit that TCP systems will not eliminate the threat posed by security flaws and hackers to computer systems. This is because the TCP standards are designed to protect unauthorised access to data, not attacks on the computer itself [Stafford, 2002a]. There have also been warnings from the groups that develop cryptographic systems that the use of encryption is not infallible, and should access be gained to the encryption keys then the security of the entire trusted system would be compromised [CNET, 2003].

The first generation of TCP systems for desktop computers will be fairly crude. The control over digital content will be carried out primarily by the application programs. However successive generations of TCP systems are likely to control content through the use of *meta data* (data about the content) that is included within the packets of files of data that are moved around computers and networks. In order to describe the content and its status all data transacted within a TCP system, and across networks between TCP systems, must be identified using a *rights expression language* (REL) [Coyle, 2004]. When a common system of RELs are agreed it will be possible to move to the next stage of DRM where the operating system relies on the meta data contained within the content, not the standards applied by the application programs which create the content, to determine how the content should be processed.

Much the debate on DRM is centred on the use of personal computers. However the area where DRM is being pioneered is in *embedded applications* – the use of computer systems to control the operation of consumer products such as mobile phones, MP3 players, DVD players, or digital audio recorders. Introducing DRM to desktop computers represents a significant increase in complexity because of the wide variety of application desktop computers perform. Therefore the first systems to introduce DRM, such as Microsoft's forthcoming *Next Generation Secure Computing Base* (codenamed *Longhorn* for short) will only implement DRM at the level of the operating system code [Microsoft, 2002]. Longhorn is due to be launched some time during 2006 [Microsoft, 2004]. Implementing a DRM system that works as part of the computer's hardware system will not happen until the introduction of the next generation of TCP-enabled microprocessors. According to the chip manufacturer Intel, this is likely to be within two or three years time. However, the work around the development of new chips for embedded applications is pioneering the development of the chip sets and code that will be required to produce DRM-compatible desktop computer systems. Even though many in the free software movement oppose DRM Linux-based systems are already involved in the development of embedded systems that use DRM [InfoWorld, 2003], and the development of Linux-based applications, so that Gnu/Linux systems will be able to implement DRM controls, is already at an advanced stage [Stafford, 2002b].

Implementing DRM Within Free Software

Linus Torvalds, who initiated the development of the Linux kernel, has expressed the view that the use of free software systems need not be in conflict with the aims of digital rights management systems [Register, 2003/LKP, 2003]. It is possible that Linus' comments merely reflect the inevitable [Cox, 2004], given the pressures from some of the largest members of the computer industry to develop trusted computer systems. For example, IBM are already working on the code that would assist the development of DRM within the Linux kernel [Stafford, 2002b]. For the free software community as a whole, Linus' announcement raises two key points:

1. Would implementing a DRM system within the Linux kernel offend the licensing provisions of the kernel and the programs that run under the kernel?
2. Would implementing DRM affect the rights of computer users to develop, modify and distribute software?

To answer the question as to whether implementing DRM using the Linux kernel would offend the Gnu *General Public License* (GPL) [FSF, 1991] we first need to define what a *Linux kernel* is. A desktop computer system runs the Linux kernel, but most of the features the computer user interacts with are provided by other programs that the kernel supports. So a *Gnu/Linux* installation on a desktop computer consists of hundreds of programs working together, under the supervision of the Linux kernel, to provide a usable computer system. Most of these programs, and the Linux kernel itself, are licensed under the Gnu GPL, but some use a variety of other software licenses.

In contrast to desktop computers, embedded applications, such as the TiVO video recorder, use a stripped-down version of the Linux kernel and custom-written, proprietary applications that implement DRM controls over the recorded data. Where a person or organisation modifies a GPL licensed program and does not redistribute it, or runs proprietary software on top of the GPL software, it is not an infringement of the Gnu GPL license [Fitzgerald, 2003]. However TiVO modified the Linux kernel in order to implement DRM within the operation of their video recorder, and so in compliance with the GPL, they released the source code for these modifications [TiVO] (consequently, people are already hacking the code to modify the operation of the video recorder [TiVO FAQ]). Therefore the development of the TiVO does not infringe the GPL as all the requirements of the license have been complied with. The only bar to the use of free software in such commercial embedded applications would be use a restrictive license that did not allow the commercial exploitation of the licensed software, for example, the Creative Commons *Non-Commercial license* [Creative Commons].

Developing a Gnu/Linux distribution that implemented DRM is more complex than developing DRM in embedded devices because it involves redistribution of the source code, and the restrictions imposed by the GPL or other open source licenses would apply to this redistribution. The fact that the code for the DRM functions has to be distributed does not affect the security of the system because the use of encryption and digital signatures ensures that each installation of the DRM code is unique. Therefore access or circumvention of the security or content protection systems involved could only take place if the attacker had access to the encryption keys controlling the digital signatures. So for those wishing to implement DRM within the Linux kernel the only consideration is, *does the Gnu GPL permit it?*

The Gnu GPL does not, as a point of deliberate design, restrict or place conditions on the re-use of code for certain uses. This non-discrimination is key to the evolution of the Gnu GPL license, and the Free Software Foundation cite such discrimination (for example licenses that do not allow the commercial use of code) as a factor in determining whether a license is free or not [FSF, 2004]. Similar criteria are used by the Open Source Institute in accrediting open software licenses [OSI, 2004a]. OSI also apply conditions relating to discrimination against software, but this only applies to the licensing of the software, not to the discrimination on the basis of 'trust' by one running program against another [OSI, 2004b].

Most of the major free and open source licenses do not preclude the introduction of DRM systems as part of the operation of the Linux kernel, or other essential programs that operate with the Linux kernel. In fact, there are already Gnu/Linux distributions available, mostly for specialist applications, that implement DRM as part of the operating system – for example, *Montavista Linux* [Montavista, 2004]. So we could view the implementation of DRM within the Linux kernel as a *fork* in development. DRM modules can be introduced under the terms of most major free and open source licenses, but precisely because of the system of open licensing these modifications cannot be enforced on all users.

DRM and the Use of Free Software

Moving to the second of the two questions posed above, would the implementation of DRM within the Linux kernel affect the ability of computer users to use, modify and redistribute software? Clearly if the licensing of free software is not a bar to the implementation of DRM, then would the free software community accept such a change? This is a more difficult question to answer because we move from the legal interpretation of software licenses into the interpretation of the social and political motives that drive the free software movement.

The initial effect of DRM on free software is likely to be a widening of the incompatibilities between the information produced by proprietary and free software. In particular, the creation of documents, images and other media from DRM-enabled applications is likely to restrict the ability of non-DRM enabled free software to interact with DRM-enabled systems. However, in the dynamic situation of electronic culture, these restrictions may further encourage the development of an alternative open media – for example *open music* or *open arts* – that could widen the amount of content currently available to the users of free software. In the longer-term, as DRM controls extend to the content servers across electronic networks and the hardware that makes these networks function, it is possible that the use of any system not compliant with the trusted computing standard could be barred. Unless non-compliant communication channels were maintained in some way, this would clearly restrict the movement of non-DRM compliant information.

Another significant issue is the effect of digital rights management controls upon *fair dealing rights*. Currently within copyright law (but varying in extent between different legal jurisdictions) the public have certain rights to breach the strict limitations of copyright. For example, the freedom to copy or quote a certain proportion of a copyrighted work for use in academic study without obtaining a licence to do so. The content industry have traditionally lobbied to restrict fair dealing. However DRM controls, enforced by trusted computing systems, have

the potential to provide absolute restrictions on the use of copyrighted works, perhaps in excess of the limits permitted to the public by fair dealing rights. How damaging this process becomes depends upon the willingness of governments to restrict the implementation of digital rights management within information systems in order to preserve fair dealing.

The other significant issue related to the development of trusted computing systems is the control of the encryption keys that the computer uses to sign and encrypt information. This has implications for the privacy of data and the security of computer systems. If a person is able to create their own digital signatures for the files they create, and is able to create their own encryption keys for the storage of data, then they are responsible for maintaining their own security. The security of the system cannot be breached unless they specifically allow it by sharing their keys. However, if TCP systems use *trusted third-parties* to produce and hold the encryption keys then the level of security and privacy the computer user has will always be dependent upon the willingness of the third party to protect the user's rights. If the third party hands over the user's keys to law enforcement agencies without first establishing the agencies legal right to take the key, or corrupt individuals within the third party's organisation divulge people's keys, then the right of the individual can be violated.

How far these trends develop depends upon how deeply the use of DRM controls extend within electronic networks. If trusted computing remains within the environment of desktop computer operating systems then the ability to use free software will be greater. However, if trusted computing grows to encompass the movement of content across networks the use of any free software, or the use of non-trusted content, would be severely restricted. Ultimately the depths to which DRM extends is likely to be dependent upon whether the public accept the potentially restrictive controls it will create, or whether discontent with the loss of functionality that the public has traditionally enjoyed leads to a backlash against the further development of DRM-enabled systems.

As outlined above, by speculating on the potential uses of the controls provided by DRM-enabled systems it is fairly easy to define a range of effects these controls might have on the computer users. However these effects are not specific to free software. They are equally valid for the effect of DRM and trusted computing restrictions within proprietary systems. There are also supporters of free source software who support DRM. To understand the motivations of those *within the free software community* who oppose DRM and trusted computing we have to understand the philosophical basis for why they advocate free software.

One of the cornerstones of the philosophy of free software is that people's use of computer systems should be unfettered by proprietary rights. The guarantee of access to the source code of any program, and the freedom to study, modify and freely distribute the source code, prevents the use of the program code from being blocked, because access to the source code allows such blocks to be circumvented.

The problem with the 'open source' model is that access to the code cannot circumvent the power of the law on intellectual property rights. One clear example of such restrictions is the use of technological standards that are patented. It is for this reason the free software community has vociferously opposed the development of software patents in the USA and Europe. However, many of the standards related to DRM are being developed by associative groups, such as the Trusted Computing Group. The terms of these technological stand-

ard are agreed between the consulting parties, like other international standards, and then adopted by them all as an open standard that supports their own models of business development [IBM, 2001]. This has two implications:

Firstly, it ensures that the standard is widely accepted and used, giving a higher chance that the standard would be adopted by the computing community as a whole. An example of such a standard would be the *hypertext transfer protocol* (HTTP – the standard that allows the operation of the World-Wide Web), developed by Tim Berners-Lee, which was deliberately released free of intellectual property claims in order to ensure its widespread adoption [Berners-Lee, 2000].

Secondly, by abandoning intellectual property claims no one organisation would have control of the standard. For this reason those implementing the standard could not be accused of operating a monopolistic practice that was likely to harm the well-being of the computer market.

If the Trusted Computing Group's terms are adopted and become dominant then the DRM standard is likely to be free. For this reason the objections from the free software community are reduced from the opposition to restrictive or proprietary standards, to a simple issue of whether DRM is good or bad for computer users. Such a judgement cannot be based upon the quality of code, or systems, but on the freedom of computer users to manipulate their equipment as they will. This argument goes to the heart of the free software debate, and divides those who work with free software for personal reasons from those who view free software merely as an effective model for the development of computer software.

Whilst the catalyst for the development of free software was undoubtedly Richard Stallman, the widespread adoption of the open development model within the mainstream computer industry has led to a diminution of the underlying motives for the development of free software. Such reification is common when the values of radical organisations are selectively adopted by mainstream society. The tactics adopted by radical social movements, described by sociological observers such as Saul Alinsky, certainly apply to the free software movement. In particular, the notion that *the goal once named cannot be countermanded* [Alinsky, 1971] applies to the aims outlined for the free software movement in Richard Stallman's *Gnu Manifesto* [Stallman, 1985]. Nearly twenty years after their release, these principles still form the core of the values promoted by the advocates of free software.

A clear change in the orientation of the free software movement came with the formation of the Open Software Initiative (OSI) in 1998. As outlined by Sam Williams in his book on Richard Stallman [Williams, 2002], the formation of the OSI was an attempt by the mainstream computer industry to make a break with the aims of the grassroots-based free software movement. In fact one of the founders, Bruce Perens, resigned shortly after the formation of OSI because it was clearly working in opposition to Stallman's objectives. This break began to reinforce the division, in Stallman's terms, between the advocates of free software and "the engineers". In fact, from the text of Linus' statement on Linux and DRM [LKP, 2003], one could almost believe that Linus sees Stallman's labelling of him as *just an engineer* as a positive, not a negative attribute.

To find a deeper meaning behind this split we can look to the divisions between the different

cultures within the computer establishment. This is perhaps most starkly defined by Pekka Himanen [Himanen, 2001]. He identifies the division in terms of a *hacker ethic* which typifies the social structures being enabled by Information Revolution, and which operates in contradiction to the *protestant work ethic* that has evolved over the Industrial Revolution. Looking at the personal motivations of those involved, this analysis certainly explains many of the differences between the vision for computing outlined by Richard Stallman and that outlined by Bill Gates [Gates, 1999]. It also explains many of the motivations for those who work voluntarily with free software, and who are seemingly unconcerned about payment, or intellectual property rights, or achieving some position within the defined structure of an organisation.

Ultimately, on Himanen's analysis, we should view the clash over the introduction of DRM into free software as an issue of personal freedom, power, control, and economic exploitation – not an issue of code, licensing and competing views of how the free software model should operate. In this sense trusted computing, and the centralisation of controls over content and the functionality of software, run wholly counter to the hacker ethic. It represents an effort by the computer establishment, still working within the economic constructs of the Industrial Age, to rein in the freedoms of communication and expression created by the *disruptive technologies* of the computer and the electronic network.

Clearly, the development of DRM controls and trusted computing systems has the potential to restrict the freedom to use information systems that has grown up with the development of the personal computer. The problem this poses, for those who see free software as the only model of software development compatible with protecting the rights of the user, is that the debate must move from what this community support (*software developed on an open model*) to a position of what they are against (*the restriction of personal freedom in order to protect the economic power of content producers*).

Conclusions

The development of DRM will have negative effects for those computer users who have traditionally enjoyed the benefits of open computing – be that open access to data or the ability to use free software. How far these negative effects develop depends upon how the technological standards around the trusted computing model are defined and implemented, and the role the computer user has in determining how the processes of trust are managed.

The conflict over DRM requires that those in the free software movement, who see trusted computing as a threat to their freedom to use computers, express this opposition in terms of *rights*, and *power*, and *economic control*. This is because the arguments that relate to open standards, or open code, or free licensing are no longer tenable as DRM and trusted computing are implementable using the free software model and open licensing system.

As free software licenses allow the development and introduction of trusted computing systems, those who oppose DRM cannot rely on the defence that it *breaks the rules*, since it does not. Likewise any argument that relates to the quality or security of free over non-free software is irrelevant, since it is this same language that the promoters of trusted computing are using to justify their development of these systems.

Therefore the opposition that the advocates of free software raise to DRM must be based upon the underlying economic and political motives of those who wish to develop these systems – the content industry, and the parts of software industry that stand to gain by the large-scale expansion of online content services that DRM would safeguard [Acquisti, 2004]. If the debate from the grass-roots side of the free software movement were to shift to this position it would break new ground because it would cause a very public polarisation between the advocates of *freedom* software and *the engineers*.

What is likely to produce such a shift is the introduction of the trusted computing standard into the Linux kernel. In addition to his statement relating to the implementation of the trusted computing standard, Linus has also made comments regarding the format of the GPL. In particular his dislike of “the politics” [eWeek, 2004a]. If those who maintain the Linux kernel do implement the trusted computing standard then we might potentially see a fork in the development of the Linux kernel. Under the terms of the Gnu GPL license the code of the non-DRM kernel would still be freely available to those wishing to maintain a non-DRM kernel, and therefore those not wishing to use DRM would be at no immediate disadvantage.

The issue that a fork would not address is the ability, against the wishes of its creators, of trusted computing and digital rights management systems to be developed under the banner of the Gnu General Public License. Not only is this permissible, but the process is already well advanced. The text of the GPL is to be revised in 2005, and some have speculated that this might lead the Free Software Foundation not to tone down, but to extend the political content of the license. The purpose of this would be to address the issues of software patents, digital rights management and trusted computing [eWeek, 2004b].

Should the text of the Gnu GPL be further radicalised then it might lead to the licensing of some free software projects, and perhaps future revisions of the Linux kernel, under licenses other than the GPL. This would represent a clear schism between the *freedom* and *engineer* wings of the free software movement. However, such a split may be beneficial for both sides as it would enable them to focus on their own priorities for the development of free software. Especially the freedom wing, which might find that the underlying political motives for the development of free software become more relevant to the public debate over computer systems as the inevitable extension of digital rights management limits society’s ability to use information systems as they would wish.

Sources

The links to the sources listed below were all accessed on the 19th December, 2004.

- Acquisti, 2004 *Darknets, DRM, and Trusted Computing: Economic Incentives for Platform Providers*, Alessandro Acquisti, Carnegie Mellon University, 2004.
<http://web.si.umich.edu/tprc/papers/2004/357/TPRC.pdf>
- Alinsky, 1971 *Rules for Radicals – A Pragmatic Primer for Realistic Radicals*, Saul D. Alinsky, Vintage, 1971.
- Anderson, 2003 *Trusted Computing and Competition Policy – Issues for Computing Professionals*, Ross Anderson, Cambridge University Computer Laboratory Security Group, published in Upgrade, vol.4 no.3, June 2003.
<http://www.upgrade-cepis.org/issues/2003/3/up4-3Anderson.pdf>
- Berners-Lee, 2000 Chapter 6, *Weaving the Web – The Past, Present and Future of the World-Wide Web, by its Inventor*, Tim Berners-Lee, Texere, 2000.
- CNET, 2003 *Trusted computing comes with a warning*, Robert Lemos, CNET News, 16th April 2003.
http://news.com.com/Trusted%2Bcomputing%2Bcomes%2Bwith%2Ba%2Bwarning/2100-1009_3-997223.html?tag=st.rn
- Cox, 2004 Email from Alan Cox (a key developer of the Linux kernel) to myself in response to some questions I posed about *Linux, Linus and DRM*, 13th December 2004.
- Coyle, 2004 *Rights Expression Languages – A report for the Library of Congress*, Karen Coyle, February 2004.
http://www.loc.gov/standards/Coylereport_finalsingle.pdf
- Creative Commons *CC Non-Commercial License*, Creative Common (undated)
<http://creativecommons.org/licenses/>
- eWeek, 2004a *Torvalds: GPL Needs Minor Work*, eWeek, 29th November 2004.
<http://www.eweek.com/article2/0,1759,1731874,00.asp>
- eWeek, 2004b *GPL 3 to Take on IP, Patents*, Peter Galli, eWeek, 22nd November 2004.
<http://www.eweek.com/article2/0,1759,1730102,00.asp>
- Fitzgerald, 2003 *Legal Issues Relating to Free and Open Source Software*, Prof. Brian Fitzgerald, Queensland University of Technology School of Law, 2003.
<http://opensource.mit.edu/papers/opensourcelawbook.pdf>
- FSF, 1991 *Gnu General Public License version 2*, Free Software Foundation 1991.
<http://www.fsf.org/licenses/gpl.html>
- FSF, 2004 *Free Software Licenses*, Free Software Foundation, 2004
<http://www.fsf.org/licenses/license-list.html>
- Gates, 1999 *Business @ The Speed of Thought*, Bill Gates, Penguin Books 1999.
- Gilmore, 2001 *What's Wrong With Coy Protection*, John Gilmore, Electronic Frontier Foundation, 16th February 2001.
<http://www.toad.com/gnu/whatswrong.html>
- Himanen, 2001 *The Hacker Ethic and the Spirit of the Information Age*, Pekka Himanen, Secker and Warburg, 2001.
- IBM, 2001 *Pervasive Computing: IBM and Open Standards*, IBM, 2001.
<http://www-306.ibm.com/software/pervasive/tech/openstand.shtml>

- Infoworld, 2003 *Linux boost expected for Trusted Computing schemes*, Paul Kill, InfoWorld, 29th January, 2003.
http://www.infoworld.com/article/03/01/29/hntcpa_1.html
- Intertrust, 2004 *About Digital Rights Management*, InterTrust Technologies Inc., 2004.
<http://www.intertrust.com/main/overview/drm.html>
- LKP, 2003 *Linus Torvalds: "DRM is Perfectly OK With Linux"*, Linux Knowledge Portal, 25th April 2003.
http://www.linux-knowledge-portal.org/en/content.php?&content/editorial/drm_torvalds.html
- Microsoft, 2002 *Microsoft 'Palladium': A Business Overview*, Microsoft, August 2002 (updated January 2003).
<http://www.microsoft.com/PressPass/features/2002/jul02/0724palladiumwp.asp>
- Microsoft, 2004 *Microsoft Announces 2006 Target Date for Broad Availability Of Windows "Longhorn" Client Operating System*, Microsoft Press Release, 2004.
<http://www.microsoft.com/presspass/press/2004/Aug04/08-27Target2006PR.asp>
- Montavista, 2004 *Montavista Linux*, Montvista Software Inc, 2004.
<http://www.mvista.com/products/>
- NYT, 2002 *Microsoft, warns 'the the era of "open computing", is ending*, New York Times, 25th July 2002.
<http://www.nytimes.com/2002/07/25/technology/25NET.html>
- OSI, 2004a *The Approved Licenses*, Open Source Institute, 2004.
<http://www.opensource.org/licenses/>
- OSI, 2004b *Clause 9, The Open Source Definition*, Open Source Institute, 2004.
<http://www.opensource.org/docs/definition.php>
- Register, 2003 *Linus Torvalds blesses DRM, and nothing happens*, The Register, Tuesday 29th April 2003.
http://www.theregister.co.uk/2003/04/29/linus_torvalds_blesses_drm/
- Stafford, 2002a *The Need for TCPA*, David Stafford, IBM, October 2002.
http://www.research.ibm.com/gsal/tcpa/why_tcpa.pdf
- Stafford, 2002b *Clarifying Misinformation on TCPA*, David Stafford, IBM, October 2002.
http://www.research.ibm.com/gsal/tcpa/tcpa_rebuttal.pdf
- Stallman, 1985 *The Gnu Manifesto*, Richard Stallman, 1985.
<http://www.gnu.org/gnu/manifesto.html>
- Stallman, 2002 *Treacherous Computing*, Richard Stallman, published in The Inquirer, 22nd October 2002.
<http://www.theinquirer.net/?article=5858>
- TiVO *TiVO – Gnu/Linux Source Code*, undated.
<http://www.tivo.com/linux/linux.asp>
- TiVO FAQ *Hacking The TiVo FAQ Wizard 1.0.3*, undated.
<http://tivo.samba.org/>
- Williams, 2002 *Free as in Freedom – Richard Stallman's Crusade for Free Software*, Sam Williams, O'Reilly, 2002.
<http://www.oreilly.com/openbook/freedom/>