UIC John Marshall Journal of Information Technology & Privacy Law

Volume 21
Issue 4 *Journal of Computer & Information Law - Summer 2003*

Article 5

Summer 2003

Locked Out: The New Hazards of Reverse Engineering, 21 J. Marshall J. Computer & Info. L. 601 (2003)

Carla Meninsky

LOCKED OUT: THE NEW HAZARDS OF REVERSE ENGINEERING

I. INTRODUCTION

In 2002, two cases were filed that rely on copyright law and the *Digital Millennium Copyright Act* ("DMCA") to prevent competition in replacement parts. One plaintiff, Lexmark, is a printer manufacturer. The other company, Chamberlain, makes garage door openers.

The DMCA prohibits users from overriding access controls to copyrighted works. While one normally thinks of the access control as protecting digital media, such as a video game, a movie or music on a CD, in these cases, the access control is protecting access to underlying hardware. Both manufacturers, in these cases, have computer chips in their appliances and replacement parts which act as a digital lock and key. The software[1] on the chips is copyrighted. The chips contain authentication sequences which prevent unauthorized access to the appliance and ensure that replacement parts come from the appliance manufacturer. In both cases, competitors have discovered, through reverse engineering, how to make replacement parts that either override or communicate with the digital lock. In DMCA terms, the competitor has circumvented a technological measure that controls access to a copyrighted work.

In the first case, Static Control reverse engineered Lexmark's lock and key mechanism to make replacement ink cartridges. In February, 2003, Lexmark, the printer manufacturer, won a preliminary injunction preventing Static Control from remanufacturing cartridges, effectively employing a "lock-out" to a competitor in the replacement part market.[2]

In the past, manufacturers have put digital locks on their devices, either through software authorization sequences, encryption keys, or various hardware devices, to prevent competitors from gaining access to their devices. The canonical example is of the video game console manufacturer who tries to lock out competing game content developers from

1. This paper uses the terms "software," "computer program" and "code" interchangeably to refer to the instructions that run on various types of hardware devices and general purpose computers.

2. *Lexmark*, 2003 U.S. Dist. LEXIS 3734 at **82-3.

using its proprietary console.[3] Typically, it does not take long for one of the competitors to reverse engineer the interface and produce its own games.

Under copyright law, reverse engineering of a product was generally upheld as fair use as long as acquisition of the product was lawful, and a new, noninfringing product was created as a result. The rationale was that there was a net benefit to society because new products promote the arts and sciences. The text of the DMCA now makes it illegal to circumvent a digital lock, and in particular, to traffic in any device that circumvents an access control. The claim of Lexmark and Chamberlain is that, by including the digital key in a competing replacement part or video game, which unlocks the digital lock in the hardware, a competitor is literally violating the DMCA. If this is true, then any new product is illegal even if the product does not infringe a copyrighted or patented work.

This paper examines how the DMCA affects the reverse engineering of replacement parts and other interoperable products. The paper starts by looking at whether reverse engineering to make such products is legal outside of the context of the DMCA. The paper then reviews alternative mechanisms a manufacturer could use to attempt to exclude competitors. The paper examines the legislative history of the DMCA to try to distill what Congress intended in enacting the DMCA. Finally, it offers possible remedies to the text of the DMCA as it currently stands. This paper contends that the DMCA chills innovation because it gives manufacturers a "monopoly" over their interfaces where they wouldn't have one otherwise.

The interface between a piece of equipment and its replacement part is usually protected solely as a trade secret. Extending copyright protection to replacement parts is a way of granting monopoly protection to a trade secret. On the other hand, if the interfacing mechanism is patented or copyrighted, this is a misuse of the legal monopoly granted to the intellectual property, by extending the monopoly to the unpatented or uncopyrighted replacement part. Without the ability to create new products that interoperate with existing devices, competitors are barred from entering the market unless they can compete on the device level. Now, only the biggest players will be able to create new products. Since the products will be, by definition, incompatible with each other, consumers will be harmed by having limited applications deriving from a single source from which to choose. Fewer innovative products will be developed.

---

3. *See generally AtariGames Corp. v. Nintendo of Am. Inc.*, 975 F.2d 832 (Fed. Cir. 1992) [hereinafter *Atari I*]; *Sega Enter. Ltd. v. Accolade, Inc.*, 977 F.2d. 1510 (9th Cir. 1992).

II. REVERSE ENGINEERING PRE-DMCA

A. Divining Trade Secrets

The Supreme Court has defined reverse engineering as "starting with the known product and working backward to divine the process which aided in its development or manufacture."[4] In the computer context, the purpose of reverse engineering is usually to create a new product that is compatible with an existing device or piece of software. Usually, the new product is not an identical copy of the reverse engineered product, but rather it is an improvement to the product or a completely new application. Because the process of reverse engineering is expensive and time consuming, companies only engage in reverse engineering when the interface to the existing technology is unavailable or maintained as a trade secret.

Reverse engineering is considered a fair, honest, and lawful means of uncovering a trade secret.[5] This notion has been incorporated into the *Uniform Trade Secret Act* ("UTSA"), the *Semiconductor Chip Protection Act*, and the *Restatement (Third) of Unfair Competition*. "Matters which are fully disclosed by a marketed product and are susceptible to 'reverse engineering' . . . cannot be protected as trade secrets."[6] Judge Posner suggests that perhaps this is because "reverse engineering involves the use of technical skills that we want to encourage."[7] The policy argument continues that reverse engineering results in the advancement of science and the arts, because understanding how a product works leads to new and improved ideas.[8] In addition, there is the general belief that "anyone should have the right to take apart and to study a product that he has [lawfully] bought."[9] This emphasizes that the acquisition of the original product must also be by fair and honest means for reverse engineering to be legal.[10]

B. Copyright Infringement and Fair Use

Since the computer reverse engineer would like his new product to interoperate with an existing device, the new product must communicate with the device in a way that the device recognizes. There are two gen-

4. *Kewanee Oil Co. v. Bicron Corp.*, 416 U.S. 470, 476 (1974).

5. *Id.*

6. *Scanvec Amiable, Ltd. v. Chang*, 2002 U.S. Dist. LEXIS 23625 at *18 (quoting *SI Handling Sys. v. Heisley*, 753 F.2d 1244, 1255 (3rd Cir. 1985)).

7. *Rockwell Graphic Sys., Inc. v. DEV Indus., Inc.*, 925 F.2d 174, 178 (7th Cir. 1991).

8. *See Bateman v. Mnemonics Inc.*, 79 F.3d 1532, 1540 n. 18 (11th Cir. 1996).

9. *Rockwell*, 925 F.2d at 178.

10. See *Restatement (Third) of Unfair Competition* § 39 cmt. f; *Atari I*, 975 F.2d at 844 (noting that in order for reverse engineering to be fair use, possession of copyrighted code must be by lawful means).

eral approaches to reverse engineering in the computer context. With the first approach, the reverse engineer analyzes how existing products communicate with the device. By capturing the input and output to the products, an engineer may be able try to isolate any commonalities or patterns that emerge, particularly at start-up time. The reverse engineer then mimics this exact sequence in his new product. The method of generating the communication sequence in the new product could be completely different from how an existing product generates it. In practice, this approach works best when the interoperability requirements are minimal. Usually, the interaction between a product and a device changes depending on the particular function and application, so interoperability requirements are difficult to derive in this manner without more information.

The second approach to reverse engineering is to examine the code that resides in the product and device themselves. The code can then be disassembled and analyzed to gain an understanding of the scope of the functionality and interface requirements. This method is difficult and time consuming particularly when the code is large. A reverse engineer may not know if particular instructions were included because of an authentication sequence, the peculiarity of the device, such as timing issues or an error in the microprocessor, an algorithm he is unfamiliar with, reservation for future expansion, obsolete code, or simple error in the code itself. He is reluctant to leave anything out that he doesn't understand for fear that it is somehow necessary to the interoperability requirements. By the same token, he is reluctant to correct any errors for fear that other software modules have compensated for a known error. Usually, the reverse engineer uses a combination of the two approaches, along with any other available material, such as user manuals, advertising, or specification sheets, which may shed light on what the device may actually be doing.

In order to reverse engineer a device's interface, it is therefore necessary to analyze the program while it is running on the device. This process creates intermediate copies of the software every time the program is run. Therefore, prior to the enactment of the DMCA, reverse engineering exposed the engineer to two potential claims of copyright infringement. The first occurred during the analysis phase. The second was in creating the software for the new product.

Courts have upheld the right to reverse engineer copyrighted software even though copies of the software are made in the process. The general belief is that one who rightfully possesses a copy of a program should be able to legally make use of it.[11] In the past, courts have called

11. Final Rep. of the Natl. Commn. on New Tech. Uses of Copy. Works 31 (1978) [hereinafter *CONTU Final Report*].

this intermediate copying fair use, as long as 1) this was done independently of insider knowledge and there was no evidence of misappropriation of trade secrets, and 2) a different product was created as a result. While there is a presumption of unfairness due to the commercial nature of reverse engineering, the public policy benefits that result from independent development of new products outweighs any negatives.[12] "[A]n attempt to monopolize the market by making it impossible for others to compete runs counter to the statutory purpose of promoting creative expression and cannot constitute a strong equitable basis for resisting the invocation of the fair use doctrine."[13] The Ninth Circuit concluded, "where disassembly is the only way to gain access to the ideas and functional elements embodied in a copyrighted computer program and where there is a legitimate reason for seeking such access, disassembly is a fair use of the copyrighted work, as a matter of law."[14]

In 2000, the Ninth Circuit continued to uphold reverse engineering as fair use. The court found that Connectix's copying of Sony's copyrighted operating system was a fair use for the purpose of gaining access to the unprotected elements of Sony's software.[15] Object code, resulting from compilation of source code, "may be copyrighted as expression, *but it also contains ideas and performs functions that are not entitled to copyright protection*."[16] Since object code "cannot . . . be read by humans," the unprotected ideas and functions must be translated in order to be discoverable.[17] Reverse engineering requires copying both the unprotected and protected elements of a copyrighted work.[18] While this involved iterative copying and use of the entire work, the final product produced by Connectix contained no infringing material from the Sony operating system.[19] Therefore, the court found that the intermediate copies were of little weight to the fair use analysis. Even though Connectix's software performed essentially the same function as Sony's, only on a different platform, the expressive element of the software was in the organization and structure of the code.[20] Therefore, the new software was transformative[21] and not infringing.[22]

12. *Sega*, 977 F.2d. at 1523.

13. *Id.* at 1523-24.

14. *Id.* at 1527-28.

15. *Sony Computer Ent., Inc. v. Connectix Corp.*, 203 F.3d 596, 602 (9th Cir. 2000) (quoting 17 U.S.C. §§ 102(a) – (b)).

16. *Id.* (emphasis in original).

17. *Id.*

18. *Id.* at 602-03 (citing *Sega*, 977 F.2d at 1518-19).

19. *Id.* at 598.

20. *Id.* at 607.

21. *Id.* at 606-07.

22. *Id.* at 608.

As with other types of expressive works, copyright infringement of software is proven if the infringer had access to the copyrighted work and the two works are substantially similar.[23] With reverse engineering, access is a given; the resulting product is due to having dismantled the original. It is the lawful acquisition of an existing product that allows a finding of permissible reverse engineering. The critical issues are, then, whether the reverse engineering was done by fair and honest means, and whether the resulting product retains appropriated copyrightable expression.

C. Protectability of Software Under Copyright

Courts consider computer programs to be basically utilitarian in function because the programs are used to operate machines to get results.[24]

The fact that a medium of expression has a functional capacity should not preclude constitutional protection. [C]omputer source code, though unintelligible to many, is the preferred method of communication among computer programmers.[25]

While the Sixth Circuit was referring to protection under the First Amendment, the sentiment applies equally well to the Copyright Clause. "[C]omputer source code is an expressive means for the exchange of information and ideas about computer programming."[26]

Therefore, it is important to distinguish between what a program does and how it does it. The expression of how to operate something does not extend copyright protection to the method of operation itself. Software to achieve a particular result may be written, or expressed, in many different ways, but to do so, it may contain many elements that are dictated by function, efficiency, standards, or by compatibility requirements.[27] Since copyright protects expressions of ideas and not the ideas themselves, computer software consists of both protectable expression and nonprotectable ideas, methods of operation and functions.

In order to deal with the complexities of software, courts have developed specific terminology for analysis of copyright infringement of software. Because there is a need to differentiate between the code itself and its output or interactions, courts talk about literal and nonliteral elements of software. "The 'literal elements' of a computer program are its source and object code. [T]he 'nonliteral elements' . . . are the products that are generated by the code's interaction with the computer hard-

23. *Computer Assoc. Intl., Inc. v. Altai Inc.*, 982 F.2d 693, 701 (2d Cir. 1992).
24. *Sega*, 977 F.2d. at 1525.
25. *Junger v. Daley*, 209 F.3d 481, 484 (6th Cir. 2000).
26. *Id.* at 485.
27. *Sega*, 977 F.2d at 1524.

ware and operating program(s)."[28] Nonliteral elements of a computer program include its parameter lists, macros, general flow charts, and user interfaces, such as screen displays and command menu hierarchies.[29]

Generally courts have found the nonliteral elements of a computer program to be ineligible for copyright protection, even in cases of literal copying.[30] The code itself, however, can be infringed by literal or nonliteral copying. In nonliteral copying, courts look for substantial similarity between the original code and the allegedly infringing code.[31] When faced with nonliteral-copying cases, courts must determine whether similarities are due merely to the fact that the two works share the same underlying idea or are trying to achieve the same result, or whether they instead indicate that the second programmer copied the first programmer's expression. The Second Circuit in *Altai* designed a test that most circuits have adopted, to deal specifically with whether one computer program copied nonliteral expression from another program's code.[32]

The *Altai* test was originally formulated "to determine whether the nonliteral *elements* of two or more computer programs are substantially similar."[33] The *Altai* court was concerned with whether there, in fact, had been unlawful copying because the two utility programs generated the same output and performed the same translation function.[34] Since there was no verbatim copying of the source or object code (which would have been literal copying of a literal element), the court looked to the structure of the two programs to decide if nonliteral copying of nonliteral

28. *Mitek Holdings Inc. v. Arce Engr. Co.*, 89 F.3d 1548, 1556 n. 15 (11th Cir. 1996).

29. *Id.* at n. 16.

30. *See e.g. Lotus Dev. Corp. v. Borland*, 49 F.3d 807, 815 (1st Cir. 1995) (holding that the Lotus menu command hierarchy is an uncopyrightable "method of operation," as used in 17 U.S.C § 102(b)); *Altai*, 982 F.2d at 716 (noting that commands generated by an interface conversion utility, which translated a request to one operating system into a request to another, was not copyrightable); *Mitek*, 89 F.3d at 1557 (stating that main menu and command tree not copyrightable "as a matter of law"); *Bateman*, 79 F.3d at 1548 n. 33 (holding that copyright protection does not extend to functional results of program execution; "such results are processes better left to patent and trade secret protection"); *Apple Computer Inc. v. Microsoft*, 35 F.3d 1435, 1439 (9th Cir. 1994) (holding that user interface elements of screen display are not copyrightable); *Gates Rubber Co. v. Bando Chem. Indus., Ltd.*, 9 F.3d 823, 842-43 (10th Cir. 1993) (noting that constants representing derivable scientific facts are not copyrightable); *Mitel, Inc. v. Iqtel, Inc.*, 124 F.3d 1366, 1373 (10th Cir. 1997) (command codes for telecommunications controller were a method of operation containing expression, however, "that expression is excluded from protection under the scenes a fair doctrine").

31. *Altai*, 982 F.2d at 706.

32. *Id.* at 711-12.

33. *Id.* at 706 (emphasis added).

34. *Id.* at 702.

elements, the program's structure and arrangement, had occurred.[35]

The *Altai* test involves three steps: abstraction, filtration, and comparison.[36] The abstraction step requires courts to start with "the allegedly copied program's structure and isolate each level of abstraction contained within it."[37] In other words, the code is broken down into its functional components according to its structure and arrangement. This step allows the protectable expression to be separated from unprotected ideas.[38] Next, courts apply a filtration step in which they examine the structural components at each level of abstraction to determine whether their particular inclusion at that level was 'idea' or was dictated by considerations of efficiency, so as to be necessarily incidental to that idea; required by factors external to the program itself; or taken from the public domain.[39]

Finally, courts compare the protected elements of the infringed work (i.e., those that survived the filtration screening) to the corresponding elements of the allegedly infringing work to determine whether there was sufficient copying of protected material to constitute infringement.[40]

In analyzing source code for similarity, standard copyright principles apply to computer software.[41] Copyright protection does not extend to algorithms and coding techniques taken from the public domain.[42] The "scenes a faire" doctrine recognizes, as stock features, industry use of standard software techniques.[43] Extrinsic factors such as hardware interfaces, compatibility and interoperability requirements, design standards, industry demands, and customary programming practices within the computer industry, are considered scenes a faire because they control the design of software.[44] Similarly, the merger doctrine recognizes that the algorithm chosen or external factors may dictate how the software is implemented.[45] To a large degree, efficiency and simplicity concerns govern how software is programmed.[46] As there are only a limited number of efficient implementations for any given task, it is likely that an efficient implementation is the result of independent creation rather than copying.[47] With embedded software, for example, the footprint of

35. *Id.*
36. *Altai*, 982 F.2d at 706-711.
37. *Id.* at 707.
38. *Id.* at 706.
39. *Id.* at 707.
40. *Id.* at 710.
41. *Altai*, 982 F.2d at 706.
42. *Id.* at 710.
43. *Id.* at 709-10.
44. *Id.*
45. *Id.* at 708.
46. *Altai*, 982 F.2d at 708.
47. *Id.*

the code is necessarily small. The code must be fast and efficient. Therefore, the closer a program approximates the simplest or most efficient form, the more likely the idea has merged with the expression.[48] If only two or three realistic alternatives exist, or an implementation is typical or obvious, the expression is said to merge into the idea.[49]

By maintaining the structure and arrangement of the code as part of the analysis, the *Altai* test is ideal for recognizing that combinations and interrelationships of functions could be protectable expression. If the program selection and arrangement is non-obvious, or consists of choices from among many options, then the structure could be protectable expression even if composed of only nonprotectable elements.[50]

With literal copying, i.e. verbatim copying of source or object code, the issue is whether the code contains copyrightable expression. A parallel type of analysis to the *Altai* test must be undertaken to filter out from comparison any unprotectable elements resulting from merger and efficiency.[51] Where literal copying is dictated by compatibility and interoperability requirements, a court may not find infringement; reverse engineers often must employ standard techniques in order for the new product to be compatible.[52]

Courts understand that a line-by-line comparison of two implementations of software may not alone reveal that literal copying has taken place. Translating a program line-by-line from one microprocessor or programming language to another is also considered literal copying. Courts find evidence of literal copying when the allegedly infringing work contains more instructions from the original work than are absolutely essential to the implementation of a particular task. For example, in *E.F. Johnson Co. v. Uniden Corp. of America*, Uniden, in creating a two-way mobile radio that was compatible with EFJ's, unwittingly included the translation of EFJ's copyright notice into its own microprocessor's language.[53] Other evidence of literal copying that the court found significant were the use of an identical programming technique when a more efficient method was available to the different microprocessor, and including the same obsolete instructions that had remained in EFJ's code from an earlier version which were no longer used.[54] The court also found that the fact that Uniden included the same errors and misunder-

48. *Id.* (citing Steven R. Englund, *Idea, Process, or Protected Expression?: Determining the Scope of Copyright Protection of the Structure of Computer Programs*, 88 Mich. L. Rev. 866, 902-03 (1990)).

49. *Id.*

50. *Softel, Inc. v. Dragon Med. & Sci. Commun., Inc.*, 118 F.3d 955, 967 (2d Cir. 1997).

51. *Bateman*, 79 F.3d at 1545.

52. *Id.* at 1547.

53. *E.F. Johnson Co. v. Uniden Corp. of Am.*, 623 F. Supp. 1485, 1492 (D. Minn. 1985).

54. *Id.* at 1495-96.

standings in the way the code was designed raised an inference of copying.[55] Since EFJ's software was copyrightable expression, and since Uniden did not use its own "imagination, creativity and independent thought" in developing its own software, the court held that Uniden had infringed EFJ's copyright.[56]

D. Digital Locks and Authentication Sequences

The source code that encrypts content has been held to be expression and therefore eligible for copyright protection.[57] However, it is the lock and key mechanism that permits the decryption that is at issue here. Since copyright protects only the expression of ideas, if encryption keys, authentication sequences, or other access mechanisms are only functional capabilities, then they should not be protectable as expression under copyright law. In the past, courts have followed this principle of treating digital locks and keys like any other software element. Courts held that the copying of the literal and nonliteral elements of an access device was legitimate as long as it was limited to only those elements essential to achieving compatibility or interoperability. "[W]hen specific instructions, even though previously copyrighted, are the only and essential means of accomplishing a given task, their later use by another will not amount to an infringement."[58]

In an infringement action, a court may approach this inquiry on either of two levels. The first focuses on whether the lock and key interaction is copyrightable. The second focuses on whether the software that generates the digital key is copyrightable and whether that has been infringed. In *Uniden*, the court found that in order to make its radios compatible with EFJ's system, Uniden was required to copy EFJ's identification sequence exactly to establish communication.[59] Therefore, EFJ's particular identification sequence was not copyrightable even though it was the result of EFJ's selection and creativity.[60] However, to create a noninfringing product, Uniden was limited to using only the specific code sequence that was essential to achieving compatibility.[61]

By the same token, in *Sega Enters., Ltd. v. Accolade, Inc.*, the Ninth Circuit found the authentication sequence to a video game console to be a purely unprotected functional element.[62] To stem software piracy, Sega

55. *Id.* at 1496.
56. *Id.* at 1502 n. 17.
57. *See Junger*, 209 F.3d at 484.
58. *CONTU Final Report*, supra n. 14, at 20.
59. *Uniden*, 623 F. Supp. at 1493-94.
60. *Id.* at 1503.
61. *Id.*
62. *Sega*, 977 F.2d at 1514.

had used its trademark as part of an authentication sequence.[63] Sega's key consisted of twenty bytes of initialization code plus the letters "S-E-G-A."[64] A game cartridge would not operate on the Sega console unless the initialization code was in a particular location in the cartridge.[65] Accolade included the verbatim sequence in its competing video games in order to achieve compatibility.[66] The court held that when there is no other method of access to the computer that is known or readily available to rival cartridge manufacturers, the use of the initialization code by a rival does not violate the [Copyright] Act even though that use triggers a misleading trademark display.[67]
Therefore, Accolade could not be prevented from using it.[68]

The question of whether unlocking a digital lock access mechanism constitutes copyright infringement has been treated similarly. The Fifth Circuit was one of the first courts to deal with this question. Vault sold a special diskette that was designed to prevent software piracy.[69] The diskette contained a digital access mechanism and a protective program.[70] The program prevented a computer from reading the content on a diskette unless the access mechanism was inserted into the computer.[71] If a user tried to copy the content onto anything other than a special Vault diskette, the Vault protective program would also be copied, and would thus prevent the content from being accessed.[72] Quaid sold diskettes that included a software program which mimicked the interaction of the Vault access mechanism with Vault's protective program.[73] When a user copied a Vault-protected diskette to a Quaid diskette, the Quaid unlocking software would permit unprotected copying.[74] Because, in the process, users copied the Vault protective software in addition to the content, Vault claimed that Quaid was liable for contributory copyright infringement.[75] The court disagreed, reasoning that the Quaid diskette permitted users to make archival copies of their own software, and the Quaid system provided users with a substantial noninfringing use.[76] Therefore, Quaid's software did not constitute contributory

63. *Id.* at 1515.
64. *Id.*
65. *Id.*
66. *Id.* at 1516.
67. *Id.* at 1514.
68. *Id.* at 1531.
69. *Vault Corp. v. Quaid Software Ltd.*, 847 F.2d 255, 256 (5th Cir. 1988).
70. *Id.* at 257 n. 1.
71. *Id.* at 256.
72. *Id.* at 256, 263.
73. *Id.* at 257.
74. *Vault*, 847 F.2d at 257.
75. *Id.* at 258.
76. *Id.* at 267.

infringement.[77]

Vault also claimed that Quaid had directly infringed its copyright. To unlock Vault's access mechanism, Quaid's initial implementation had included 30 characters of Vault's source code.[78] The court found that this sequence was a "quantitatively minor amount" of code when compared to the overall 50 pages of source code.[79] The remainder of Quaid's code was otherwise not substantially similar.[80] Vault argued that this sequence, however, was qualitatively significant, because it "constituted the identifying portion" which unlocked the access mechanism.[81] However, the court looked at the transformative nature of Quaid's program. Instead of performing what it saw as an identical *locking* function, the court saw Quaid's software as performing an *unlocking* function.[82] Since the two were fundamentally opposing applications, the court held that they were not qualitatively similar.[83]

While this reasoning may have shown a lack of understanding by the court of the underlying functionality (the Vault program, in addition to preventing access, also had to have been able to unlock the access mechanism), it does fit in with the reverse engineering doctrine. Quaid had reverse engineered the mechanism through lawful acquisition. It had copied only the essential sequence necessary to interact with Vault's digital lock and the rest of its code had been developed without regard to Vault's software.

In *Atari Games Corp. v. Nintendo of America, Inc.*, the access mechanism was a successful comparison of two signal streams that were independently generated by the console and the cartridge. The issue was whether the output stream sent from the cartridge to the console was copyrightable.[84] The signal stream was produced by a predetermined random seed at startup. Because the numbers were arbitrary rather than the result of specific choices, the court held that they did not meet the originality requirement of *Feist* and did not merit copyright protection.[85]

The court reasoned that copyright protection was available for a program's output only where the expression of the output itself was a proper

77. *Id.*
78. *Id.* at 257.
79. *Vault*, 847 F.2d at 267.
80. *Id.*
81. *Id.* at 268.
82. *Id.*
83. *Id.*
84. *Atari Games Corp. v. Nintendo of Am., Inc.*, 1993 U.S. Dist. LEXIS 8183 at *6 (N.D. Cal. 1993) (on remand) [hereinafter *Atari II*].
85. *Id.* at **12-14 (referring to *Feist Publications v. Rural Tel. Serv. Co.*, 499. U.S. 340 (1991) (holding a minimal amount of creativity is necessary for CR protection; a work is eligible for protection if it is the result of specific choices and arrangement).

subject for copyright. For example, when a program generated an audiovisual display, the display could be worthy of copyright protection as an audiovisual work. Nor could copyright protection be extended to the particular timing of the sequence, because timing was merely "the process by which electronic signals [were] created, transmitted, and received."[86] The copyright laws explicitly exclude protection for any "process, system [or] method of operation."[87] The court held that a competitor may copy those portions of the program which were necessary to have the cartridge chip send the proper sequence of bits at the proper time to the console chip, and may include those portions in the final version of the program. However, Nintendo could prove infringement by showing that Atari had copied more than was necessary to produce the authentication sequence.

Therefore, a lock and key were held to be purely functional, nonliteral elements of a digital system. Courts ferreted out the boundaries of the locking mechanism to determine what was sufficient to achieve interoperability. Literal copying within those bounds was permitted even when what was being copied was a protected trademark or otherwise would have been copyrightable, because of its nonliteral status in the system.

But, reverse engineering prior to the DMCA came with its own hazards. It was imperative to distill only the essential elements from a copyrighted work for access and compatibility. The more complex or arcane the code, the more difficult that task was. If the original work was large, the time and investment could be substantial. A manufacturer, thus, could recoup his investment during the time it took a competitor to divine his trade secret, and society benefited from new and innovative products.

E. Alternative Lock-Out Mechanisms

1. *Patent Protection*

Manufacturers have other means available to prevent competitors from developing compatible products. An effective way to lock out competitors is to patent the lock and key mechanism. If an access control is patented, a reverse engineer may explore the limits of the locking mechanism under the experimental use doctrine, but he may not subsequently embed the digital key of a patented access control in a marketed product. Moreover, the provisions of 35 U.S.C. § 271(d) provide the patentee with limited power to exclude others from competition in non-staple goods. A non-staple good is a component or related article that has no use or purpose other than to work in conjunction with the patented article. The

86. *Id.* at **17-18.
87. 17 U.S.C. § 102(b) (2003).

patentee is thus able "to eliminate competitors and thereby to control the market" for the non-staple good as well as his patented article.[88] A seller may be a contributory infringer under § 271(c) "if he makes a non-staple article that he knows was 'especially made or especially adapted for use'" with a patented article.[89] For contributory infringement to apply, the users of the article must be liable for direct infringement.[90] There is a repair exception to § 271 which allows a competitor to manufacture a "replacement [to] a spent part of a combination patent, which is not separately patented."[91] By patenting the lock and key components separately from the device or software they are protecting, this exception does not come into play.

An example of this is demonstrated in *Atari II*. Nintendo, as well as copyrighting its authentication program, patented its digital security system. The patent claimed the separate digital lock and key devices which held the stored authenticating programs, as well as the mechanism for unlocking the console based on the results of the authentication sequence. The court found that Atari had infringed the Nintendo patent by including an unlocking device—the digital key—in its game cartridges. The court noted that this was contributory infringement under § 271(c), rather than direct infringement, because Atari's use only included the second half of the authentication system, and Atari had not identified any uses for its device other than in connection with Nintendo's console.[92]

Other lock and key mechanisms have been successfully patented.[93] In fact, Lexmark had patented its toner cartridge chip technology, indicating in its patent disclosure that it was an effective way to lock-out competitors. Patent protection makes it possible for a digital key to bar competitor access to a proprietary interface. The drawback is that patent protection is not available to a nonoriginal access control mechanism. Thus, a manufacturer is forced to either innovate an original solution to access control or license an existing solution. Nonetheless, this is a reasonable accommondation in promoting the advancement of science and the arts. If the lock is that innovative, it deserves the rewards of patent protection. Patent protection thus provides an incentive to the manufac-

88. *Dawson Chem. Co. v. Rohm & Hass*, 448 U.S. 176, 230-31 (1980).

89. *Husky Injection Molding Sys. v. R&D Tool & Engr. Co.*, 291 F.3d 780, 784 (Fed. Cir. 2002) (citing 35 U.S.C. § 271 (1994)).

90. *Id.*

91. *Id.* at 786 (quoting *Aro Mfg. Co. v. Convertible Top Replacement Co.*, 365 U.S. 336, 345 (1961)).

92. *Atari II*, 1993 U.S. Dist. LEXIS at ** 51-52.

93. *See e.g. Rackman v. Nintendo*, 1994 U.S. Dist LEXIS 16931 (1994) (holding valid a patented access mechanism which employed public and private key cryptography; the cartridge would not function if the program on it had not been properly encrypted).

turer who desires to be the exclusive owner of all the components of his device.

2. *Contracts and Licensing Agreements*

Another means of excluding competitors is through contract law. To protect proprietary interfaces when computers or appliances are sold, the software that runs on them is usually transferred under restrictive licensing agreements. These agreements attempt to prevent reverse engineering and disclosure of source code to third parties. Oftentimes, the agreements bind the licensee to exclusively deal with the manufacturer for maintenance and replacement. The Federal Circuit has consistently upheld restrictive licensing agreements.

In *Bowers v. Baystate Technologies*, the Federal Circuit held that the substantial similarity between two software products was evidence that Baystate had violated a shrink-wrap license that prohibited any reverse engineering.[94] The similarities "extended beyond structure and design to include many idiosyncratic design choices and inadvertent design flaws."[95] In deciding the issue, the court found that the majority of courts have held that "the Copyright Act does not preempt contractual constraints on copyrighted articles."[96]

In an earlier case, *DSC Communications Corp. v. Pulse Communications, Inc.*, the Federal Circuit concluded that a contract in which the manufacturer specifically retained title to copies of licensed software and limited the right to transfer copies or disclose details of the software to third parties, was inconsistent with the rights of owners of copies of software.[97] Therefore, the licensees, as nonowners, were limited in their rights with regard to the software copies.[98] DSC manufactured telecommunications switching systems.[99] Both DSC and Pulse, a competitor, manufactured interface cards which plugged into DSC's backplane.[100] On power up, a copy of the DSC operating software was downloaded into

94. *Bowers v. Baystate Tech.*, 302 F.3d 1334, 1343 (Fed. Cir. 2002).

95. *Id.*

96. *Id.* at 1342; *see e.g. ProCD, Inc. v. Zeidenberg*, 86 F.3d 1447, 39 U.S.P.Q.2d 1161 (7th Cir. 1996) (holding that a shrink-wrap license was not preempted by federal copyright law); *Wrench LLC v. Taco Bell Corp.*, 256 F.3d 446, 457 (6th Cir. 2001) (holding a state law contract claim not preempted by federal copyright law); *Nat'l Car Rental Sys., Inc. v. Computer Assocs. Int'l, Inc.*, 991 F.2d 426, 433 (8th Cir. 1993); *Taquino v. Teledyne Monarch Rubber*, 893 F.2d 1488, 1501 (5th Cir. 1990); *Acorn Structures v. Swantz*, 846 F.2d 923, 926 (4th Cir. 1988); *but see Lipscher v. LRP Publs., Inc.*, 266 F.3d 1305, 1312 (11th Cir. 2001).

97. *DSC Commun. Corp. v. Pulse Commun., Inc.*, 170 F.3d 1354, 1361 (Fed. Cir. 1999) [hereinafter *DSC*].

98. *Id.* at 1361-62.

99. *Id.* at 1357.

100. *Id.* at 1358.

the RAM of every interface card.[101] Because, the telephone companies were only licensees, rather than owners, the Federal Circuit found that downloading a copy of DSC's software onto a Pulse card infringed DSC's copyright.[102] Section 117 of the *Copyright Act* only allows "owners" the privilege of copying software onto a machine for operation purposes.[103] Pulse was therefore liable for contributory infringement.[104]

In order to be compatible, Pulse had reverse engineered DSC's interface.[105] On power-up, the operating system checked two locations on the interface card for authentication purposes.[106] If the card was not authenticated, the interface card did not work.[107] Pulse had carefully copied only the essential elements of DSC's software to achieve interoperability.[108] However, because Pulse had used a telephone company lab to detect the power up sequence, the Federal Circuit held that the telephone company may have violated its licensing agreement.[109] If that was so, then Pulse would have obtained the boot codes through unfair and dishonest means.[110]

However, not all courts find restrictive licensing agreements valid. A minority of courts hold that licenses forbidding reverse engineering are an attempt to avoid the first sale doctrine.[111] A California court recently held that a license that consists of a single payment for an unlimited term of possession is, in fact, a sale to a customer.[112] Clauses in such agreements forbidding reverse engineering were therefore void. Many countries outside the United States also do not uphold licensing provisions which forbid reverse engineering.[113] Sklyarov, a Russian researcher who worked for Elcomsoft, disclosed an eBook security flaw at a security conference. Johansen, a Norwegian resident and citizen, had reverse engineered the CSS copy protection program for DVDs and developed DeCSS.[114] Both exploited the same flaw in public key encryption which enabled them to extract the keys from the content rather than

101. *Id.*

102. *DSC*, 170 F.3d at 1362.

103. 17 U.S.C. § 117 (2003).

104. *Pulse*, 170 F.3d at 1362.

105. *See id.* at 1363-64.

106. *Id.* at 1364.

107. *Id.*

108. *Id.*

109. *Pulse*, 170 F.3d at 1364.

110. *Id.* at 1364-65.

111. *Softman Prod. v. Adobe*, 171 F. Supp. 2d 1075, 1084 (N.D. Cal. 2001) (citing *Novell, Inc. v. CPU Distrib., Inc.*, 2000 U.S. Dist. LEXIS 9975 at *18 (S.D. Tex. 2000)).

112. *Id.* at 1086.

113. *See e.g. DVD Copy Control Assoc. v. Bunner*, 113 Cal. Rptr. 2d 338 (2001); *Elcom*, 203 F. Supp. 2d 1111.

114. *Universal City Studios, Inc. v. Reimerdes*, 111 F. Supp. 2d 294, 311 (2000).

directly decrypt the content themselves. Both Johansen and Sklyarov allegedly violated click-wrap license agreements which prohibited reverse engineering. However, their respective countries held such license agreements invalid, rendering their actions lawful in their countries.

In a suit brought under the California UTSA, DVD-CCA sued several Web site operators for posting DeCSS, asserting that CSS was its protectable trade secret.[115] Since California's UTSA recognized reverse engineering as a "proper means" for obtaining a trade secret, the court held that "the only way in which reverse engineering could be considered improper means would be if whoever did the reverse engineering was subject to the click [wrap] license agreement."[116] Declining to interpret Norwegian law to decide if Johansen's reverse engineering was lawful, the case was decided under First Amendment grounds.[117]

F. Antitrust Considerations

Competitors, who have been locked out of the replacement part market, have brought suit under the *Sherman Act* for antitrust violations. However, such suits have not typically been successful in proving that the appliance manufacturer has an illegal monopoly. The crucial question in any monopoly claim is determining the relevant market. That market depends on whether the replacement part is considered a separate product from the appliance.[118] If there are two products, then the question is whether the manufacturer has engaged in unlawful tying or exclusive dealing in the replacement part market.[119]

When identifying whether tying is in issue, the Supreme Court has held that "the answer to the question whether one or two products are involved turns not on the functional relation between them, but rather on the character of the demand for the two items."[120] A tying arrangement cannot exist unless there is a sufficient consumer demand for the purchase of the tied product separate from the purchase of the tying product.[121] Tying arrangements are illegal only if they force purchases that would not otherwise be made.[122] Thus, the seller must have market power in the tying market and that is where the Court focuses its inquiry.[123]

---

115. *Bunner*, 113 Cal. Rptr. 2d at 341.

116. *Id.* at 344.

117. *Id.* at 351.

118. *Eastman Kodak Co. v. Image Technical Serv., Inc.*, 504 U.S. 451, 462 (1992) (Scalia, J. dissenting).

119. *Id.*

120. *Jefferson Parish Hosp. Dist. No. 2 v. Hyde*, 466 U.S. 2, 19 (1984).

121. *Id.* at 21-22.

122. *Id.* at 27.

123. *Id.* at 18.

It is not unusual for an appliance manufacturer to have a natural monopoly over its own replacement parts.[124] In *Kodak*, the question was whether Kodak had used this monopoly power to unlawfully achieve a monopoly over service for its equipment. The Supreme Court found that Kodak had engaged in tying.[125] In the first part of its analysis, the Court determined that service and parts were separate markets. While an entire industry had developed around the providing of service to owners of photocopying machines, there remained sufficient demand for replacement parts from self-service owners.[126] Kodak's sale of parts to third parties on condition that they buy service from Kodak was unquestionably a tying arrangement.[127] The critical issue, therefore, was whether the tying was illegal under the *Sherman Act*.

Kodak's position was that the arrangement did not violate the antitrust laws because its market power over its replacement parts was constrained by its position in the interbrand photocopier market. It claimed that competition from other manufacturers limited the prices that Kodak could charge for its service and parts. Nevertheless, the court held, "[t]he fact that the equipment market imposes a restraint on prices in the after-markets by no means disproves the existence of power in those markets."[128] The Court found that for the service-parts market to affect equipment demand, customers must be able to determine the total cost of the equipment over the lifetime of that purchase.[129] However, when the cost of obtaining information is high, customers do not engage in this form of life-cycle pricing.[130] Moreover, once a customer has invested in a particular brand of equipment, if the cost of switching to another brand is high, the customer will feel locked in and tolerate some amount of overpricing.[131] In this case, there was evidence that the cost of switching to a competitor's copier was very high and that Kodak varied its price to customers based on the package sold.[132] In denying summary judgment, the Court concluded that, in order to prevail, Kodak needed to prove that its parts, service and equipment were one unified product that were controlled by the equipment market.[133]

Kodak has had a major effect on subsequent allegations of monopoly or tying in derivative markets. In a suit against a different competitor,

124. *Kodak*, 504 U.S. at 489-90 (Scalia, J. dissenting).
125. *Id.* at 464.
126. *Id.* at 463.
127. *Id.*
128. *Id.* at 471.
129. *Kodak*, 504 U.S. at 473.
130. *Id.*
131. *Id.* at 476.
132. *Id.* at 477.
133. *Id.* at 486.

DSC filed suit against DGI for manufacturing interface cards which plugged into its telecommunications switching backplane.[134] As described earlier, the software that ran on DSC's switching system controlled all of the interface cards that were plugged into its backplane. The software module that allowed the interface cards to communicate with the switching system software was downloaded onto each interface card at power up. DGI's replacement/expansion cards also worked by accepting DSC's downloaded software module. DGI reverse engineered the authentication process that allowed the download. DSC sued DGI for copyright infringement and misappropriation of trade secrets. DGI counterclaimed alleging that DSC violated § 2 of the *Sherman Act*.

The Fifth Circuit dismissed DGI's antitrust counterclaim against DSC. The court looked at *Kodak* to conclude that the relevant market was not the replacement part market, but the larger equipment market because its customers engaged in "life-cycle" pricing when they purchased the original equipment. There were many competing products in the equipment market and there was no evidence that this particular manufacturer had any market power in the larger market. The fact that they had a monopoly in their replacement part market (replacement parts were not interchangeable in the larger market) did not seem to matter. The Sixth Circuit narrowed the *Kodak* holding even further, holding that antitrust concerns are raised only if the manufacturer changes its policy after customers are locked in and switching costs to an alternative solution are too high.[135]

Therefore, if an appliance manufacturer can claim life-cycle pricing, then the analysis changes to one product rather than two. The appropriate market segment for monopoly purposes changes from the replacement part market to the appliance market. In the appliance market, usually there are many competitors. In today's pro-consumer environment, consumers are informed about average costs and frequency of repair of major purchases. Comparative shopping guides, such as Consumer Reports, allow many products, such as major appliances and motor vehicles, to be bought based on life-cycle pricing. In order to prevail in an antitrust claim, a competitor would have to show the appliance manufacturer had substantial market control in the more general appliance market. Threat of an antitrust suit is not a realistic means of gaining aftermarket access.

134. *Alcatel USA, Inc. v. DGI Technologies, Inc.*, 180 F.3d 267 (5th Cir. 1999).
135. *PSI Repair Serv. v. Honeywell*, 104 F.3d 811, 820 (6th Cir. 1997).

III. REVERSE ENGINEERING UNDER THE DMCA

A. THE NEW HAZARDS

Companies employ reverse engineering when it is necessary to discover the interoperability requirements for a compatible product. If the interface that is being derived contains an access control or digital lock, the DMCA is implicated at three stages of the reverse engineering process. The first time is during the actual reverse engineering phase when the engineer analyzes how the existing product works and determines what measures are needed to unlock the digital lock and what the requirements are for achieving compatibility. The second is when the reverse engineer develops a digital key that works with the existing product. The third context is when the reverse engineer embeds this digital key into his new product, so that each time the new product is used, it is able to unlock the underlying program or device. Otherwise, it will not be able to interoperate with the device. The DMCA may exempt the first two activities. The third stage is the most problematic because it exposes the reverse engineer to the anti-trafficking provisions.

The DMCA states that "[n]o person shall circumvent a technological measure that effectively controls access to a [copyrighted] work."[136] In order for this clause to be invoked, there must be a lock on the device that controls access to the device, and the underlying device or program must be copyrightable. In addition, the DMCA has three provisions which guard against trafficking in circumvention technology. Two are of interest. Section 1201(a)(2)(A) forbids anyone from "manufactur[ing]. . .or otherwise traffic[king] in any technology. . .that is primarily designed or produced for the purpose of circumventing" an access control.[137] Section 1201(a)(2)(B) forbids trafficking in a circumvention device whose only purpose is to circumvent a digital lock. It would seem that the provisions would not be an issue in a typical reverse engineering case since the digital key which performs the circumvention is usually only ancillary to the new product that is created, rather than being the primary purpose of the reverse engineering. However, if a court focuses solely on the digital key component, rather than the complete product that results from the reverse engineering process, the key itself could be seen as violating all three of the DMCA trafficking provisions. The key's sole purpose is to circumvent an access control. By selling a product that includes the key, one is "trafficking" in circumvention technology.

The DMCA contains an exemption for reverse engineering as long as the reverse engineering does not itself constitute copyright infringement or violate some other law. Section 1201(f) permits a reverse engineer to

136. 17 U.S.C. § 1201(a)(1)(A).
137. *Id.* § 1201(a)(2)(A).

circumvent a digital lock for the sole purpose of identifying those elements of a computer program that are necessary to achieve interoperability with an "independently created computer program."[138] Interoperability is defined as information exchange between two programs.[139] The engineer may develop a compatible digital key and use it in order to analyze the interoperability requirements.[140] Section 1201(f)(3) limits the use of that digital key; it may be made available to others only if its sole purpose is to achieve interoperability with the new program. While the text of the DMCA implies that the reverse engineer may validly embed the developed key into his new product and then offer the combination for sale, if a court does focus only on the component key, this limitation could potentially expose the reverse engineer to the anti-trafficking provisions. Also, the reverse engineer must be careful not to infringe any existing copyright, otherwise the exemption does not apply at all.[141]

One court that has construed the text of the reverse engineering provisions ruled that the language of § 1201(f)(3) permits only the person who performed the reverse engineering to make the resulting informa-

138. *Id.* § 1201(f)(1).

> *Id.* § 1201(f) Reverse engineering. (1) Notwithstanding the provisions of subsection (a)(1)(A), a person who has lawfully obtained the right to use a copy of a computer program may circumvent a technological measure that effectively controls access to a particular portion of that program for the sole purpose of identifying and analyzing those elements of the program that are necessary to achieve interoperability of an independently created computer program with other programs, and that have not previously been readily available to the person engaging in the circumvention, to the extent any such acts of identification and analysis do not constitute infringement under this title.
>
> (2) Notwithstanding the provisions of subsections (a)(2) and (b), a person may develop and employ technological means to circumvent a technological measure, or to circumvent protection afforded by a technological measure, in order to enable the identification and analysis under paragraph (1), or for the purpose of enabling interoperability of an independently created computer program with other programs, if such means are necessary to achieve such interoperability, to the extent that doing so does not constitute infringement under this title.
>
> (3) The information acquired through the acts permitted under paragraph (1), and the means permitted under paragraph (2), may be made available to others if the person referred to in paragraph (1) or (2), as the case may be, provides such information or means solely for the purpose of enabling interoperability of an independently created computer program with other programs, and to the extent that doing so does not constitute infringement under this title or violate applicable law other than this section.
>
> (4) For purposes of this subsection, the term "interoperability" means the ability of computer programs to exchange information, and of such programs mutually to use the information which has been exchanged.

139. *Id.* § 1201(f)(4).

140. *Id.* § 1201(f)(2).

141. *Id.*

tion available to others.[142] The Web site operator, who posted the descrambing software, DeCSS, to the Internet, had not personally performed the reverse engineering, even though he had acquired the software from the person who did.[143] Nor did the court credit the developer's testimony that the software was developed to achieve interoperability with a different operating system.[144] While the court was responding to a digital key which had been trafficked separately from its original intended use, this has direct implications on a business who uses independent distributors and dealers to sell products that contain a reverse engineered digital key. The court left unanswered the question whether this logic applied to reverse engineered digital keys that are embedded in a competing product.

The reverse engineering exemption is very narrow. The text specifies only program to program interoperability.[145] Digital locks and keys in hardware may not come under this exemption. Many authentication sequences consist only of data exchanges. Courts construing encryption schemes, that make use of public and private keys, may view the keys as data and not program interoperability. These issues have yet to be tested in the courts.

Most DMCA cases to date have been challenges to tools or devices that allow users to circumvent copy controls or other use restrictions that protect the exclusive rights of the copyright owner. Selling or offering such tools to the public is a violation of the DMCA.[146] The purpose of the digital locks in these cases is to protect digital content from piracy. Because the goal of the reverse engineer is to achieve interoperability, the reasoning in these cases should not apply. The reverse engineer in the interoperable context is creating his own competing content, rather than pirating existing content. Nonetheless, the DMCA does not make this distinction. Circumvention of a copy control is usually accomplished through the reverse engineering of the copy control. Rather than focus on the purpose of the reverse engineering, the DMCA only distinguishes between access controls and copy controls. Under the DMCA, circum-

142. *Universal City Studios, Inc. v. Reimerdes*, 111 F. Supp. 2d 294, 320 (S.D.N.Y. 2000).

143. *Id.*

144. *Id.*

145. *See* 17 U.S.C. § 1201(f)(1) (stating that it is "necessary to achieve interoperability of an independently created computer program with other programs"); 17 U.S.C. § 1201(f)(2) (favoring "the purpose of enabling interoperability of an independently created computer program with other programs"); 17 U.S.C. § 1201(f)(3) (noting that "solely for the purpose of enabling interoperability of an independently created computer program with other programs"); 17 U.S.C. § 1201(f)(4) (stating "the ability of computer programs to exchange information").

146. *Id.* § 1201(b).

venting a copy control is always permissible, whereas, circumvention of an access control is only permissible under a few limited exemptions.

In one of the first cases to test the access mechanism provisions of the DMCA, the court found that a circumvention device violated the DMCA despite the fact that no copyright could be infringed in using it.[147] Nor were pirated copies of copyrighted works being sold as a result. SCEA claimed that GameMasters was engaging in contributory infringement for selling a Game Enhancer card which allowed US owners to play imported Sony games.[148] The court disagreed.[149] A consumer's choice to play an imported, authentic Sony game cannot be infringing Sony's copyright since the games were legally manufactured and sold in Japan.[150] The games did not become illegal bootlegs simply by being imported into the U.S.[151] However, the court held GameMasters to be in violation of § 1201(a)(2)(A) because the Game Enhancer card circumvented an access control that normally prevented the games from playing.[152]

The DMCA also exposes reverse engineers to potential criminal prosecution. DMCA imposes criminal sanctions for trafficking in circumvention devices.[153] For criminal liability to apply, the DMCA requires the willful trafficking of circumvention devices for financial gain. In the DeCSS cases, the alleged violators had merely made the circumvention techniques available to anyone on the Internet. Because there was no direct financial gain, the criminal provision did not apply. Nor would encryption researchers, another DMCA exemption, be prosecuted for merely sharing information about results of circumvention research.[154] The purpose of reverse engineering, on the other hand, is primarily for financial gain. The question is whether "willful" means knowing that you have engaged in the act of trafficking or whether knowing that the act of trafficking in a particular device is a violation of § 1201. Under copyright law, criminal sanctions apply only when someone knows that their acts are infringing.[155] "'Willful' means a 'voluntary, intentional violation of a known legal duty.'"[156]

147. *Sony Computer Ent. Am., Inc. v. GameMasters, Inc.*, 87 F. Supp. 2d 976, 986 (N.D. Cal. 1999).

148. *Id.* at 985-86.

149. *Id.* at 987.

150. *Id.* at 986.

151. *Id.*

152. *Sony*, 87 F. Supp. at 987-88.

153. 17 U.S.C. § 1204.

154. *Id.* § 1201(g).

155. *See U.S. v. Cross*, 816 F. 2d 297, 300 (7th Cir. 1987); *U.S. v. Wise*, 550 F.2d 1180, 1195 (9th Cir. 1977), *U.S. v. Moran*, 757 F. Supp. 1046, 1049 (D. Neb. 1991).

156. *Moran*, 757 F. Supp. at 1049 (quoting *Cheek v. United States*, 498 U.S. 192, 200 (1991)).

B. CONSTITUTIONALITY OF THE DMCA

The courts, which have examined it, have upheld the DMCA to be constitutional. The *Reimerdes* court held that, while computer software was expression protected by the First Amendment, the DMCA targeted only the functional components of software and was therefore content-neutral.[157] Content neutral restrictions are upheld "if they serve a substantial government interest and restrict First Amendment freedoms no more than necessary."[158] The court found that the anti-trafficking provisions furthered the important government interest of protecting copyrighted digital media from piracy, and, at the same time, were no broader than necessary to achieve the goals of "preventing infringement and promoting the availability of content in digital form."[159] Nor must regulations "be the least speech-restrictive means of advancing the Government's interest."[160]

On appeal, Corley argued that the DMCA, by preventing the dissemination of DeCSS, violated his First Amendment rights.[161] Upholding the lower court's findings, the *Corley* court affirmed that the DMCA was a necessary application of Congress' "practical policy judgments."[162] In the digital context, preventing the dissemination of computer programs capable of bypassing access controls, was the only way to stem the "virtually unstoppable infringement of copyright."[163] Since the prohibition against posting such software to the Internet targeted only the non-speech, functional component of the software, no First Amendment rights were abridged.

Corley also challenged the DMCA for restricting his right of fair use under the Copyright Clause and the First Amendment.[164] He called on the court to interpret the DMCA narrowly to avoid Constitutional collisions.[165] The court found no such collisions. It held that the language of the DMCA does not regulate the *use* of the copyrighted material, but "simply clarifies that the DMCA targets the *circumvention* of digital walls."[166] The court then stated that fair use was not constitutionally required.[167] Nor had fair use ever been "held to be a guarantee of access to copyrighted material in order to copy it by the fair user's preferred

157. *Reimerdes*, 111 F. Supp. 2d at 327, 329.
158. *Id.* at 327-28.
159. *Id.* at 330.
160. *Id.* (quoting *Turner Broad. Sys, Inc. v. FCC*, 512 U.S. 622, 662 n. 201).
161. *Universal City Studios, Inc. v. Corley*, 273, F.3d 429, 436 (2d Cir. 2001).
162. *Id.* at 452.
163. *Id.*
164. *Id.* at 436.
165. *Id.* at 443.
166. *Id.* (emphasis added).
167. *Id.* at 458.

technique or in the format of the original."[168]

In the other major case to have considered the constitutionality of the DMCA, Elcomsoft challenged the constitutionality of the DMCA on multiple grounds.[169] The first was that the DMCA was unconstitutionally vague and therefore violated the due process clause of the Fifth Amendment.[170] The second was that the DMCA violates the First Amendment because, among other things, "it impermissibly infringes upon the First Amendment rights of third parties to engage in fair use."[171] Elcomsoft developed and sold a product which allowed purchasers of Adobe eBooks to override the use restrictions on the eBooks they now lawfully owned.[172] The product enabled an owner "to engage in 'fair use' of an eBook without infringing the copyright laws."[173] However, because the owner could then distribute unlawful copies, the product also permitted the owner to engage in copyright infringement. Elcomsoft was charged with violating DMCA § 1201(b).[174]

In response to the first challenge the court held that the text of the DMCA expressly states the prohibition on trafficking applies to all circumvention tools, not just those which solely allow infringement.[175] There was no exception for tools which permitted fair use.[176] Therefore, the DMCA was not unconstitutionally vague.[177] Under its fair use analysis, the court found that Congress' purpose in enacting the DMCA was to "promot[e] the continued growth and development of electronic commerce. . . and [to protect] intellectual property rights."[178] The court recognized that the DMCA made fair use more difficult for the average user who could not develop circumvention tools, but held that Congress decided that protection against piracy was the more compelling interest.[179] Because the DMCA does not ban the circumvention of use restrictions, the DMCA still preserves a lawful owner's right to fair use of the copyrighted work.[180]

In considering whether the DMCA's restrictions were overbroad, the courts solely focused on the compelling government interest in controlling digital content piracy. This reasoning is not applicable to restric-

168. *Id.* at 459.
169. *U.S. v. Elcom*, 203 F. Supp. 2d 1111, 1122 (N.D. Cal. 2002).
170. *Id.*
171. *Id.*
172. *Id.* at 1118.
173. *Id.*
174. Elcom, 203 F. Supp. 2d at 1119.
175. *Id.* at 1124.
176. *Id.*
177. *Id.* at 1125.
178. *Id.* at 1129 (quoting H.R. Rpt. 105-551, pt. 2, at 23, Burton Decl. Ex. O) (1998).
179. *Elcom*, 203 F. Supp. 2d at 1131.
180. *Id.*

tions prohibiting the use of information as the result of reverse engineering. Moreover, in determining the constitutionality of the DMCA, the *Corley* and *Reimerdes* courts' judgments were colored because the Web site operators themselves did not claim to be making fair use of any copyrighted materials.[181] Nor was there any evidence as to the impact of the anticircumvention provisions on prospective fair-users.[182] The court simply balanced the equities before it and found there to be significant potential harm to the digital content industry. In a footnote, the *Corley* court recognized that while there may be alternative means of prohibiting unauthorized access to copyrighted materials, the defendants had the burden of proof in showing that either such technology existed or an alternate scheme, such as royalties, were effective.[183] The ultimate choice belonged to Congress.

IV. THE DMCA AND IP MISUSE

A. CONSTRUING THE DMCA IN PRACTICE

In order for the DMCA to be invoked, the digital lock, must be protecting a validly copyrighted work. Yet, the access device with which the reverse engineer is typically faced, is protecting access to the underlying hardware. In the case of a video game, the digital lock is controlling access to the game console. If a game cartridge does not have the proper digital key, the console will not unlock and allow the game to play. In a case such as *Lexmark*, the digital lock is controlling access to the printer. If the refill cartridge does not contain the proper digital key, the hardware chip on the printer will not recognize that a new ink cartridge has been installed, thereby preventing the printer from printing. The correct focus then is whether the video game console or the printer is a validly copyrighted work, or more narrowly, whether the authentication process used in the access mechanism has copyrightable elements. Since courts have previously found the digital locks and keys themselves to be functional and not protectable under copyright, in order to prevail on a DMCA claim, a manufacturer must be able to point to some expression beyond the lock and key which is copyrightable. If the lock is being used to extend a manufacturer's control beyond the copyrighted expression to a replacement part or device interface, then that is IP misuse.

B. LOCKS AS MISUSE DEVICES

"[A] patent owner may not take the property right granted by a patent and use it to extend his power in the marketplace improperly, i.e.

181. *Corley*, 273 F.3d at 459.
182. *Id.*
183. *Id.* at 456 n. 28.

beyond the limits of what Congress intended to give in the patent laws."[184] Similarly, copyright misuse is a defense that prohibits a copyright owner from extending the limited monopoly that is granted to him by the Copyright Office.[185] "The misuse defense prevents copyright holders from leveraging their limited monopoly to allow them control of areas outside the monopoly."[186] A defendant in a copyright infringement suit need not prove an antitrust violation to prevail on a copyright misuse defense.[187] A finding of misuse is thus not based on the market share of the copyright holder, but in finding that the copyright holder is using his copyright as the mechanism to gain exclusive rights over a noncopyrighted work.[188]

For example, in *Alcatel*, while DGI was unable to prove that DSC had monopoly power over its market, DGI was able to prove copyright misuse.[189] DSC sold its equipment to customers, but the software than ran on the equipment was merely licensed to the customers.[190] The software licensing agreement authorized the software to be used only with DSC's switching equipment and interface cards.[191] The court found that DSC's customers would violate their licensing agreements if they used another manufacturer's replacement parts.[192] The court held that this was copyright misuse because it allowed DSC to gain commercial control over its uncopyrighted or unpatented interface cards, "thereby securing for DSC a limited monopoly" over its replacement parts.[193]

The courts that have recognized copyright misuse have been in cases where there were "improper attempts to enlarge a copyright monopoly through restricted or exclusive licensing."[194] In *Practice Mgmt. Info. Corp. v. AMA*, the Ninth Circuit ruled that the AMA misused its copyright when it licensed its medical procedure coding scheme in exchange for an agreement not to use a competing system.[195] This exclusivity requirement gave the AMA a substantial and unfair advantage over its competitors. By agreeing to license the [coding scheme] in this manner,

184. *Atari I*, 897 F.2d at 1576; *see generally* Dan L. Burk, *Anti-Circumvention Misuse* (Public Law and Legal Theory Research Paper Series, Research Paper No. 02-10, 2002) (discussing the historical basis for the misuse doctrine).

185. *A&M Records, Inc. v. Napster, Inc.*, 239 F.3d 1004, 1026 (9th Cir. 2001) (citing *Lasercomb Am., Inc. v. Reynolds*, 911 F.2d 970, 977-79 (4th Cir. 1990)).

186. *Id.*

187. *Alcatel*, 166 F.3d at 799.

188. *Id.*

189. *Id.*

190. *Id.* at 777.

191. *Id.*

192. *Id.* at 793.

193. *Id.* at 794.

194. *A&M Records, Inc. v. Napster, Inc.*, 114 F. Supp. 2d 896, 923 (N.D. Cal. 2000).

195. 121 F.3d 516, 520 (9th Cir. 1997).

the AMA used its copyright 'in a manner violative of the public policy embodied in the grant of a copyright.'[196]

A copyright owner has the right to exclude anyone from using his work. However, if an interface to a device is a functional, nonprotectable element, then enforcing a digital lock protecting that interface extends a copyright holder's exclusive rights beyond the valid limits of his copyright. This is copyright misuse. In the case of a replacement part, the copyright holder's monopoly is extended to cover the uncopyrighted replacement part. In the case of a video game console, the monopoly is extended to all competing videogames. In the case of an operating system, the monopoly is extended to all competitors' application software. The DMCA is thus an extremely powerful tool to prohibit competition if courts focus on circumvention and trafficking instead of the purpose of the reverse engineering.

C. *Lexmark* v. *Static Control*

Lexmark, is a perfect example of the power of the DMCA as a anticompetitive weapon. Static Control reverse engineered the Lexmark printer and cartridge software. The cartridge chip contained a 37-byte program which informed the printer when the ink was low in the cartridge.[197] By watching the communication between the two chips, Static Control believed the digital key for Lexmark's digital lock consisted of the printer chip receiving the proper "ink full" message from a replacement cartridge. Static Control remanufactured spent cartridges by refilling the cartridges with ink and inserting its own chip that generated Lexmark's "ink full" message. It thought the 37 bytes were the authentication sequence. In reality, seven bytes in specific memory locations on the Lexmark cartridge chip served as the authentication sequence.[198] Thus, Static Control included more of Lexmark's cartridge software than was absolutely essential to execute a digital key. The court acknowledged that:

it would be extraordinarily difficult to determine the existence and location of [the values] on Lexmark's microchips without any contextual information to assist in determining the meaning and significance of the bytes on the microchips.[199]

Nonetheless, the court found that Static Control infringed Lexmark's copyright.[200] The court held that the 37 bytes were, in their

196. *Id.* at 521 (quoting *Lasercomb Am., Inc. v. Reynolds*, 911 F.2d 970, 977 (4th Cir. 1990)).

197. *Lexmark*, 2003 U.S. Dist. LEXIS 3734 at *11.

198. *Id.* at *12.

199. *Id.* at *25.

200. *Id.* at *35.

entirety, copyrightable expression.[201] The program was Lexmark's best approximation of when a customer should replace an ink cartridge, not merely measurable facts and formulas.[202] Disregarding arguments of merger and programming efficiency,[203] the court found that the program could have been written in a number of different ways.[204] According to the court, even if the entire program was a lock-out code, Static Control's implementation would be infringing unless it contained only those similarities that were necessary to produce the sequence that would unlock the copyright owner's lock.[205] "Public policy favors requiring competitors to carefully study security systems and discern what is truly necessary for compatibility."[206]

The court found that Static Control engaged in "wholesale, identical copying . . . for commercial exploitation and profit," thereby prohibiting a finding of fair use.[207] The court then denied a finding of copyright misuse, ignoring the fact that the copyrighted program was being used to obtain a monopoly over the ink cartridge replacements.[208] Instead, the court narrowly focused on the 37-byte program, finding that Lexmark's lock solely protected the printer and cartridge software.[209] "[A]n infringement claim [brought] against a party that has engaged in wholesale copying. . .cannot be considered misuse."[210] "Lexmark's efforts to enforce the rights conferred to it under the DMCA cannot be considered an unlawful act undertaken to stifle competition."[211] In a footnote citing *PSI Repair*, the court dismissed Static Control's antitrust claims, stating that Lexmark's policies have been in existence, and unchanged, for many years.[212]

The court next turned to the DMCA, stating that, because previous courts had held the text of the DMCA to be unambiguous, it was inappropriate to consider any policy arguments or legislative history.[213] Following the holding of *GameMasters*, the court held that the protections of the DMCA "were never intended to be limited" to protection against piracy of digital content.[214] The court found that Static Control violated

201. *Lexmark*, 2003 U.S. Dist. LEXIS 3734 at **48-52.
202. *Id.* at **50-51.
203. *Id.* at *14.
204. *Id.* at **48-49.
205. *Id.* at *40.
206. *Lexmark*, 2003 U.S. Dist. LEXIS 3734 at *41 (referring to *Atari I*, 975 F.2d at 843).
207. *Id.* at *39.
208. *Id.* at *59.
209. *Id.*
210. *Id.* at **59-60.
211. *Lexmark*, 2003 U.S. Dist. LEXIS 3734 at *60.
212. *Id.* at *59 n. 3; *see* discussion *supra* Part IV.C.
213. *Id.* at *62.
214. *Id.* at *68.

each of the three anti-trafficking provisions for both Lexmark's printer and cartridge programs.[215] Lexmark's authentication sequence was a digital lock because it controlled a consumer's ability to make use of the printer and cartridge.[216] By mimicking the sequence, Static Control's key unlocked the lock. Static Control's key was designed and marketed for no other purpose than to circumvent the lock.[217] The DMCA exemption for reverse engineering did not apply, according to the court, because Static Control's key was not an independently created program, nor did it solely interoperate with an independently created program.[218] In addition, because Static Control infringed Lexmark's copyright, under § 1201(f)(3), the reverse engineering exemption did not apply.[219]

By defining the boundary of the lock to be the seven bytes, rather than the 37-byte cartridge program, Lexmark achieved two critical advantages. First it allowed Lexmark to show that Static Control had taken more than what was essential to achieving compatibility. If all 37 bytes were required to unlock the digital lock, Static Control's use of the program would not have been infringing. However, in declining to consider efficiency, the court ignored the very real possibility that Lexmark's cartridge code was a merger of idea and expression. If there were only one or two ways to implement that expression, given the 37-byte space limitation and the industry standard of efficiency, Lexmark's code would not be copyrightable, even if, as the court held, Lexmark's code was the result of unique choice and expression.

Second and most important for the DMCA, in defining the bounds of the lock as they did, Lexmark was able to narrow the focus of the court to be solely on its software, instead of viewing the software as merely an embedded component in controlling the operation of a printer. Static Control believed it had incorporated a key which was only ancillary to the new product created. The primary product was the refilled ink cartridge. The cartridge software had no separate commercial value other than as an operating component. The cartridge and ink are merely staple articles unprotected by patent or copyright. Lexmark was able to use the DMCA, not only to extend its patent and copyright monopolies to replacement cartridges, but to the sale of replacement ink as well. This is IP misuse made permissible by the DMCA.

215. *Id.* at **65-66.

216. *Lexmark*, 2003 U.S. Dist. LEXIS 3734 at *28.

217. *Id.* at **27-28.

218. *Id.* at *73.

219. *Id.*

D. Legislative History and Intent of the DMCA

The court in *Elcom* found that the intent of Congress in passing the DMCA was to protect copyrighted works from piracy. "[O]nly regulation of the devices by which [content] is delivered will successfully save . . . intellectual property rights."[220] Looking at the legislative history, it is clear that Congress did not intend to lock out reverse engineering of devices. Congress was solely concerned about piracy of creative content. "Title I of this bill . . .creates the legal platform for launching the global digital on- line marketplace for copyrighted works."[221] The legislative history indicates that Congress expressly enacted the DMCA based on its authority under the Commerce Clause rather than its authority under the Copyright Clause.[222] The access provisions were included because Congress saw circumventing an access control like breaking into a library to gain access to a copyrighted book.[223] However, the anticircumvention provisions did not just refer to software access controls. In particular, Congress was concerned with hardware devices, such as "black-boxes," which allowed widescale piracy.[224] The DMCA was not aimed at devices, such as VCR's and personal computers which had commercially significant noninfringing uses.[225]

Congress may extend copyright-like protection under other Constitutional provisions to works which may not otherwise meet the requirements of the Copyright Clause. If the statute passed by Congress is not fundamentally inconsistent with the Copyright Clause and is otherwise within Congress' Commerce power to enact, then the statute is not an unconstitutional exercise of congressional power. On the other hand, if the statute is irreconcilably inconsistent with a requirement of another constitutional provision, then the enactment exceeds congressional authority.[226] Thus, in order to determine if Congress has the authority to extend copyright-like protection to interfaces and access control mechanisms under its Commerce power, it is necessary to examine whether the DMCA is consistent with the Copyright Clause. The *Elcom* court warned that the Copyright clause has a limited times requirement and Congress would be prohibited from "conferring intellectual property rights of per-

220. *Elcom*, 203 F. Supp. 2d at 1140.

221. Sen. Rept. 105-190, at II Leg. History (1998).

222. H.R. Rept. 105-551 (II), at Title I, § 107(d) (1998).

223. *Id.* (I), at Chap. 12 § 1201(e).

224. S. Rept. 105-190, at V. Section – by – Section Analysis / Title I. Wipo Treaties Implementation § 103.

225. 144 Cong. Rec. H7094 (Aug. 4, 1998); H.R. Rept. 105-551 (I) (noting that "[i]t is drafted carefully to target 'black boxes,' and to ensure that legitimate multipurpose devices can continue to be made and sold"). *Id.*

226. *Elcom*, 203 F. Supp. 2d at 1139-40 (citing *U.S. v. Moghadam*, 175 F.3d 1269, 1280-81 (11th Cir. 1999)).

petual duration" under the Commerce Clause.[227] The *Corley* court recognized that this was an issue under the DMCA, but declined to consider the question before it was ripe.[228]

When Congress was drafting the DMCA, the computing industry was concerned that the restrictions on circumvention were overbroad. Over several days of Congressional hearings, they urged Congress to change the language of the trafficking provisions to be inclusive rather than exclusive.[229] Trafficking in circumvention technology should only be unlawful if the technology was primarily designed for circumvention and had no other commercial purpose, and the developer had to have knowledge that it was being marketed for circumvention purposes. They asked for "a provision to ensure that the prohibition on circumvention does not limit the ability to decompile computer programs to the extent permitted currently under the doctrine of fair use."[230]

The computing industry was afraid that Congress, [b]y focusing on the technological act of circumvention in and of itself, as opposed to copyright infringement, the [DMCA] creates a number of problems, among them the significant diminution of fair use. If the new legislation does not use copyright as the criterion for violation of the copyright act, then fair use is not a limitation on liability.[231]

Indeed, they were correct. The *Corley* court held that, since the DMCA was not a copyright provision, fair use was not a defense to the DMCA. The DMCA does not concern itself with the use of copyrighted materials after the circumvention occurs.[232]

However, the only hazards Congress took note of, were those in forcing VCR and DVD manufacturers to continually update their machines with the latest authentication mechanisms created by the content industry.[233] In particular, Congress was concerned that embedded watermarks would visually degrade content.[234] Solely focused on protecting digital content, Congress did not understand that the reverse engineer-

227. *Id.* at 1141 n. 8.

228. *Corley*, 273 F.3d at 445.

229. H.R. Subcomm. On Courts & Intellectual Property of the Comm. On the Judiciary, *Copyright Treaties Implementation Act, Hearings on H.R. 2280, 2281,* 105th Cong. (June 05, 1998) (statement of Chris Byrne, Director of Intellectual Property, Silicon Graphics, Inc.).

230. 144 Cong. Rec. at H7097.

231. H.R. Subcomm. On Courts & Intellectual Property of the Comm. On the Judiciary, *Copyright Treaties Implementation Act, Hearings on H.R. 2280, 2281,* 105th Cong. (Sept. 17, 1997) (statement of Edward J. Black, President, Computer & Communications Industry Assn.).

232. *Corley*, 273 F.3d at 443.

233. 144 Cong. Rec. at H7101.

234. H.R. Rept. 105-551 (II), at Title II § 202.

ing was not an end in and of itself, but that a product was then created which would necessarily include what was learned in the process.

Congress believed it was preserving the development of new technology. The reverse engineering provisions were explicitly included to allow

> legitimate software developers to continue engaging in certain activities for the purpose of achieving interoperability to the extent permitted by law prior to the enactment of this chapter. The objective is to ensure that the effect of current case law interpreting the Copyright Act is not changed by enactment of this legislation for certain acts of identification and analysis done in respect of computer programs. *See, Sega Enterprises Ltd. v Accolade, Inc.*, 977 F.2d 1510, 24 U.S.P.Q.2d 1561 (9th Cir. 1992). The purpose of this section is to foster competition and innovation in the computer and software industry.[235]

Congress wanted to make sure that § 1201 would not be asserted against persons engaged in reverse engineering of copyrighted works and inserted the exemptions so that the DMCA would not "constitute a serious impediment to the development and production of competitive goods and services."[236]

V. PROPOSED REVISIONS TO THE DMCA

Wording specific exemptions to the provisions of the DMCA is a difficult task. The method of protecting digital media from piracy can be identical to the method of excluding competitors from a device's interface. By targeting the access mechanism rather than copyright protection, the DMCA makes it hard to distinguish what the lock is protecting.

There are different approaches for remedying the DMCA. Congress could promulgate legislation that targeted copyright violations instead of access mechanisms, or it could decide to define specific exemptions to the current DMCA that focused on the use of the lock. Until Congress does revise the language of the DMCA, courts could choose to interpret the DMCA in a way that would prohibit manufacturers from profiting from IP misuse. Just as courts developed the concept of IP misuse in the patent and copyright contexts, a similar equitable doctrine could be easily extended to the DMCA context. A court would first define the boundaries of the locking mechanism to determine what was sufficient to achieve interoperability. Literal copying within those bounds would be permitted if the lock was, either, controlling access to a connection or interface necessary to achieve interoperability or compatibility with a new product, program or device; or controlling replacement of a nonpatented staple good. Although a specific activity might come within the

235. Sen. Rept. 105-190.

236. 144 Cong. Rec. at H7079.

literal text of the DMCA, courts could fashion a rule that would narrow its interpretation.

Alternatively, the DMCA itself has provisions for regulatory revision by the Copyright Office.[237]

[D]uring each succeeding 3-year period, the Librarian of Congress. . .shall make the determination in a rulemaking proceeding. . .of whether persons who are users of a copyrighted work are, or are likely to be. . .adversely affected in their ability to make noninfringing uses. . .of copyrighted works.[238]

During the recent 2002-2003 rulemaking, the Copyright Office has granted Static Control's petition to consider whether embedded software is subject to the DMCA.[239]

Not surprisingly, Static Control proposes a special exemption for embedded printer software.[240] In addition, the petition proposes two more general exemptions. The first exempts "computer programs embedded in a machine or product and which cannot be copied during the ordinary use of the machine or product."[241] This is too broad an exemption because it would allow any digital key to be copied regardless of what it is protecting. For example, this exemption would permit DVD copy controls to be overridden. The exemption must target the manufacturer that is using the lock to extend its control over a nonpatented, staple good.

Static Control's next proposed exemption does a better job of making a distinction between the kinds of products that should be protected by locks and those that should be exempted. This proposes to exempt computer programs embedded in a machine or product and that control the operation of the machine or product connected thereto but that do not otherwise control the performance, display or reproduction of copyrighted works that have an independent economic significance.[242]

It is a good start in permitting competitors to develop replacements parts, however, the language is not comprehensive enough. It does not actually solve Static Control's case where the lock was an exchange of data. The exemption needs to cover hardware and data locks as well as computer programs. The exemption should read,

> any access mechanism, whether it be a computer program that controls the operation of the machine or product connected thereto, a physical hardware device, specific data, encryption/decryption key, or other

237. 17 U.S.C. § 1201(a)(1)(C).

238. *Id.*

239. *Lexmark*, 2003 U.S. Dist. LEXIS 3734 at *29.

240. Petition of Static Control Components, Inc., For Consideration of New Information, No. RM 2002-4 (Jan 23, 2003).

241. *Id.*

242. *Id.*

> method, that is embedded in a machine or product, but that does not otherwise control the performance, display or reproduction of copyrighted works that have an independent economic significance.

But that is not all. Under the *Lexmark* holding, the current language of the DMCA would have precluded Accolade from making competing video games. The lock on Sega's console prevented a competitor's game, an audiovisual work, from being displayed or performed. The above exemption does not solve this problem. There must be a DMCA exemption if a manufacturer is using a lock to extend its control over a nonpatented interface or connection, if the lock prevents a competitor from achieving compatibility or interoperability with any other computer program or device. An interface is defined broadly to be any connection necessary to achieve interoperability. To give an example of the different possible elements of an interface, the following is a nonexclusive list:

1) any application, library or system programming interface or protocol, including macros and subroutine calls;
2) any accompanying data and timing;
3) any nonliteral program element, such as command lists or menus;
4) any hardware register, input/output queue, memory;
5) any sensor or wireless connector; and
6) any physical connector.

The exemption should read:

> any access mechanism, whether it be a computer program that controls the operation of the machine or product connected thereto, a physical hardware device, specific data, encryption/decryption key, or other method, that controls the ability to connect or interoperate with a device, program or interface that is necessary to achieve interoperability or compatibility with a new product, program or device.

To effectively give a competitor the ability to create a compatible or interoperable product, any exemption must target IP misuse.

VI. CONCLUSION

A competitor engages in reverse engineering when it is impossible to discover by other means the interoperability requirements for a compatible product. If access to the interface is controlled by a digital lock, the reverse engineer may be violating the DMCA by embedding the digital key into his new product. Each time the product is used, the key will unlock the lock. If a court views the key as a separate component from the final product, selling a product that contains the embedded key exposes the reverse engineer and his dealers to potential civil and criminal liability.

The DMCA may only be invoked when the lock controls access to a copyrightable work. To determine if a violation has occurred, courts look at whether the authentication process used in the access mechanism is

copyrightable, and whether the reverse engineer has infringed that copyright. While no court has been willing to make a per se rule, courts have generally held that computer interfaces are ineligible for copyright protection. Use of the interfaces is necessary to achieve compatibility. Nor have courts found authentication sequences for digital locks to be copyrightable. Therefore, before the DMCA was enacted, a reverse engineer could create a digital key by copying an exact authentication sequence, and embed it in his competing product, without infringing the copyright of the owner of the digital key. The only caveat was that the reverse engineer could only take the elements from a copyrighted work that were essential for access and compatibility. But the tradeoff was fair. It gave a manufacturer ample time to recoup his investment, while providing innovators the opportunity to compete.

Interoperability is best served without the DMCA restrictions on circumventing access controls. The tests designed by the courts for permissible embedding of a digital key, are extremely comprehensive and already very narrowly tailored to detect infringement. Even without the DMCA, a court would most likely have found Static Control's implementation to be infringing. Issues of merger aside, reverse engineers are forced to take only what is needed to make a compatible product. Static Control's literal copying of Lexmark's code would only have been appropriate if the entire 37-byte program had been the key.

In deciding on the constitutionality of the DMCA, courts weighed the anticircumvention restrictions only in relation to the government interest of protecting copyrighted digital media from piracy. No court looked to see what affect the provisions would have in preventing access to the device or interface itself. At the time, no one fully understood the impact that the restrictions could have on the development of new technology. While Congress' intent was to prevent the rampant spread of digital piracy, at the same time, Congress believed it was preserving the development of new technology and the practice of developing innovative products through reverse engineering. However, the digital locks in these cases are not protecting digital content from piracy, but preventing a reverse engineer from achieving interoperability. By holding that the DMCA was no broader than necessary to promote the availability of digital content, the unintended consequence has been to give manufacturers an invincible tool of exclusion. One court has even held a circumvention device to violate the DMCA despite the fact that no copyright could be infringed in using it. The DMCA is thus demonstrably broader than necessary to achieve its goals.

The purpose of the digital locks should determine the criteria for infringement. The reverse engineer who circumvents a lock to achieve interoperability for his competing content should be exempted from the DMCA's anticircumvention and antitrafficking provisions. The focus

should be on what the lock is protecting, not the lock itself. A lock is purely functional, even if the implementation of the lock is thought to be "expression." If the purpose of the lock is to protect digital content from piracy or other copyright infringement, breaking or circumventing the lock should be illegal. In *Lexmark*, however, the purpose of the lock, was to keep the printer from printing. It was not to protect the ink cartridge's 37-byte program from rampant piracy. Congress or the Copyright Office need to fashion exemptions which combat such blatant IP misuse. In the interim, courts should be attuned to the possibility of misuse and narrow the reach of the DMCA in those situations.

The DMCA is legalizing copyright misuse by giving unlimited term protection to an interface which is not protectable under copyright. The DMCA was never meant to lock-out competition. Nor was it meant to give Copyright-like protection to trade secrets. Since there is no time limit to the DMCA's protection, it in fact gives a monopoly for an infinite term. As technology lasts at most a few years, this does not seem like much of a threat. Nonetheless, it permanently locks out competitors from providing new and improved products for those devices. While in the replacement part context the loss may be a potential price break or improved service to consumers, in other contexts, this chills innovation. Without the ability to create new products that interoperate with existing devices, small innovators are barred from entering the market unless they can compete on the device level. Most cannot afford to develop and market competing hardware. Only the biggest players will be able to create new products. Since the products will be, by definition, incompatible with each other, consumers will be harmed by having limited applications deriving from a single source from which to choose. Competition brings greater productivity, creativity, and efficiency to existing products. Society benefits as a result.

Carla Meninsky†

† J.D. Candidate, 2004, The George Washington Law School; B.A., 1977, Stanford University. Former software consultant and President of RLO Consulting, Inc., specializing in computer graphics and multimedia technology. The author would like to thank Professors Robert Brauneis, Roger E. Schechter and Thomas D. Morgan for their support in preparing this paper.