

As of Dec 3, 2001

THE LAW & ECONOMICS OF REVERSE ENGINEERING

by
Pamela Samuelson* & Suzanne Scotchmer**

- I. Introduction
- II. Reverse Engineering in Traditional Manufacturing Industries
 - A. A Legal Perspective on Reverse Engineering
 - B. An Economic Perspective on Reverse Engineering
 - C. Anti-plug Mold Laws: An Exception to Reverse Engineering Rules?
- III. Reverse Engineering In the Semiconductor Industry
 - A. Perturbations in Product Life Cycles in the Chip Industry
 - B. Copyright or Sui Generis Protection for Chip Designs?
 - C. Economic Rationale for the SCPA Rules
 - D. Post-SCPA Developments
- IV. Reverse Engineering in the Computer Software Industry
 - A. Reverse Engineering of Software And Copyright Law
 - B. The Economics of Interoperability
 - C. Reverse Engineering of Software And Contract Law
- V. Reverse Engineering of Technically Protected Digital Content
 - A. Emerging Markets in Technically Protected Works
 - B. Circumstances Leading Up To the DMCA Anti-Circumvention Rules
 - C. An Economic Analysis of the DMCA Rules
- VI. Reverse Engineering as a Policy Lever
 - A. Ways to Regulate Reverse Engineering
 1. Regulating a Market-Destructive Means of Reverse Engineering
 2. A Breadth Requirement For Products of Reverse Engineering

* Professor of Law and Information Management, University of California at Berkeley. I wish to thank Kirsten Roe, Eddan Katz, and Christine Duh for their excellent research assistance in connection with this article. Research support for this paper was provided by NSF Grant No. SES 9979852. We are both grateful for insightful comments on earlier drafts by Rochelle Cooper Dreyfuss, Joseph Farrell, Neil Gandal, Robert J. Glushko, Wendy J. Gordon, Mark A. Lemley, Stephen Maurer, Ejan MacKaay, David McGowan, Michael Moradzadeh, Maureen O'Rourke, Eva Oglieska, Jerry Reichman, Dan Rubinfeld, Hal Varian, Fred von Lohmann, and participants in the Boston University Intellectual Property Workshop Series, the University of Washington Law School's Innovation Workshop, and the Yale Legal Theory Workshop.

** Professor of Economics and Public Policy, University of California at Berkeley. Research support for this paper was provided by NSF Grant No. 98 18689.

3. Purpose- And Necessity-Based Criteria for Determining the Legitimacy of Reverse Engineering
4. Regulating Reverse Engineering Tools
5. Restricting Publication of Information Discovered By a Reverse Engineer

B. Policy Options When Innovators Try to Bypass Reverse Engineering

1. Avoiding the Threat of Reverse Engineering By Contract
2. Avoiding the Threat of Reverse Engineering by Technical Obfuscation

VII. Conclusion

I. Introduction

Reverse engineering has a long history as an accepted practice. What it means, broadly speaking, is the process of extracting know-how or knowledge from a human-made artifact.¹ Lawyers and economists have endorsed reverse engineering as an appropriate way to obtain such information, even if the intention is to make a product that will draw customers from the maker of the reverse-engineered product.² Given this acceptance, it may be surprising that reverse engineering has been somewhat under siege in the past few decades.

While some encroachments on the right to reverse engineer have been explicit in legal rule-making, others seem implicit in new legal rules that are altogether silent on reverse engineering, including the Agreement on Trade-Related Aspects of Intellectual Property Rights (TRIPS)³ and the Economic Espionage Act of 1996 (EEA).⁴ TRIPS is

¹ This is a broader definition than has previously been used by courts and commentators, but captures how the term is used in this paper. “Human-made artifacts” refers to objects that embody knowledge or know-how previously discovered by other people. Hence, the engineering required to uncover the knowledge is “reverse” engineering. As we shall see, extraction of this knowledge can be costly or cheap, time-consuming or fast, depending on the artifact, and these notions govern the consequences of allowing it to be extracted. The standard legal definition, from *Kewanee Oil Co. v. Bicron Corp.*, 416 U.S. 470, 476 (1974), is “starting with the known product and working backwards to divine the process which aided in its development or manufacture.” Professor Reichman conceives of this knowledge as applied scientific or industrial know-how. See, e.g., J.H. Reichman, *Computer Programs as Applied Scientific Know-How: Implications of Copyright Protection for Commercialized University Research*, 42 Vand. L. Rev. 639, 656-62 (1989). Treatise author James Pooley has emphasized that the “fundamental purpose of reverse engineering is discovery, albeit of a path already taken.” See JAMES H.A. POOLEY, *TRADE SECRET LAW* sec. 5.02 at 5-19 (1999). All of these notions fit within our simple notion of extracting knowledge from a human artifact.

² See, e.g., James Pooley, *supra* note 1 at 5-16; David Friedman, William M. Landes, & Richard A. Posner, *Some Economics of Trade Secret Law*, 5 J. Econ. Persp. 61, 71 (1991). See also sources cited *infra* notes 26-36 and 162.

³ See Final Act Embodying the Results of the Uruguay Round of Multilateral Trade Negotiations, Apr. 15, 1994, reprinted in *THE RESULTS OF THE URUGUAY ROUND OF MULTILATERAL TRADE NEGOTIATIONS--THE LEGAL TEXTS 2-3* (Gatt Secretariat ed. 1994); Marrakesh Agreement Establishing the World Trade Organization, Apr. 15, 1994, Annex 1C: Agreement on Trade-Related Aspects of Intellectual Property Rights (hereinafter “TRIPS Agreement”), reprinted in *RESULTS OF URUGUAY ROUND*, *supra* at 6-19, 365-403. The trade secrecy provision of the TRIPS Agreement is

an international treaty which, among other things, obligates member states of the World Trade Organization to protect trade secrets, yet it neither requires nor sanctions a reverse engineering privilege.⁵ The EEA created the first federal cause of action for trade secrecy misappropriation. Its lack of a reverse engineering defense has troubled some commentators because rights granted under the EEA arguably implicate certain reverse engineering activities previously thought to be lawful.⁶

Among the explicit legal challenges to reverse engineering are these: In the 1970's and 1980's some states forbade the use of a direct molding process to reverse engineer boat hulls.⁷ In the late 1970's and early 1980's, the semiconductor industry sought and obtained legislation to protect chip layouts from reverse engineering to make clone chips.⁸ In the mid-1980's and early 1990's, a controversy broke out about whether decompilation, a common form of reverse engineering of computer programs, was legal as a matter of copyright law.⁹ Even after U.S. courts ruled that decompilation was acceptable purposes such as achieving interoperability,¹⁰ a related controversy broke out as to the enforceability of licenses forbidding reverse engineering of software and other digital information.¹¹ More recently, questions have arisen about whether decompilation of computer programs infringes patent rights in software components.¹² In 1998, Congress outlawed reverse engineering of technical protections for digital versions of copyrighted works; this law also outlaws manufacture or distribution of tools for such

Article 39. For Congressional approval of the TRIPS and WTO Agreements, see Uruguay Round Agreements Act, Pub. L. No. 103-465, §§101-103, 108 Stat. 4809 (1994).

⁴ Pub. L. No. 104-294, 110 Stat. 3488 (codified at 18 U.S.C. ss. 1831-1839).

⁵ See TRIPS, supra note 3, art. 39. But see Charles R. McManis, *Taking TRIPS on the Information Superhighway: International Intellectual Property Protection And Emerging Computer Technology*, 41 Vill. L. Rev. 207 (1996)(arguing that reverse engineering of software is acceptable within the TRIPS framework).

⁶ The "troubling" absence of a specific reverse engineering privilege in the EEA has been noted in James H.A. Pooley, Mark A. Lemley, & Peter J. Toren, *Understanding the Economic Espionage Act of 1996*, 5 Tex. Intell. Prop. L.J. 177, 195 (1997). See also Rochelle C. Dreyfuss, *Trade Secrets: How Well Should We Be Able to Hide Them? The Economic Espionage Act of 1996*, 9 Fordham Intell. Prop., Media, & Ent. L.J. 1, 15 (1998). Specifically, the concern is that decompilation and disassembly of computer programs, which are now considered to be fair means of obtaining trade secret information in programs, may run afoul of the new EEA rules which forbid duplicating trade secrets. See Pooley et al., supra, at 195-96. See also Craig L. Uhrich, *The Economic Espionage Act: Reverse Engineering and the Intellectual Property Public Policy*, 7 Mich. Telecomm. Tech. L. Rev. 147 (2001)(recommending amendments to the EEA to privilege legitimate reverse engineering activities).

⁷ See Paul Heald, *Federal Intellectual Property Law and the Economics of Preemption*, 76 Iowa L. Rev. 959, 960 (1991). Anti-plug mold laws are discussed infra Section II-C.

⁸ See Section III.

⁹ Decompilation transforms machine-readable electronic impulses of object code into human-readable form. See Section IV-A.

¹⁰ See, e.g., *Sega Enterp. Ltd. v. Accolade, Inc.*, 977 F.2d 1510 (9th Cir. 1992), discussed in Section IV-A.

¹¹ See Section IV-D.

¹² See, e.g., Julie E. Cohen and Mark A. Lemley, *Patent Scope and Innovation in the Software Industry*, 89 Calif. L. Rev. 1, 6 (2000), discussed in Section IV-C.

reverse engineering (except in very limited circumstances) and outlaws disclosure of information obtained in the course of lawful reverse engineering.¹³

Our objectives in this article are, first, to review important legal developments regarding the right to reverse engineer, and second, to understand their economic consequences.

We start in Section II with a discussion of the well-established legal right to reverse engineer manufactured goods. In our view, the legal rule favoring reverse engineering in the traditional manufacturing economy has been economically sound because reverse engineering is generally either costly, time-consuming, or both. Either costliness or delay can protect the first comer enough to recoup his initial research and development (R&D) expenditures.¹⁴ If reverse engineering (and importantly, the consequent reimplementations) of manufactured goods becomes too cheap or easy, as with plug-molding of boat hulls, it may be economically sound to restrict this activity to some degree.

In Sections III, IV and V, we consider the law and economics of reverse engineering in three information-based industries: the semiconductor chip industry, the computer software industry, and the emerging market in technically protected entertainment products, such as DVD movies. In all three contexts, rules restricting reverse engineering have been adopted or proposed. We think it is no coincidence that proposals to restrict reverse engineering have been so common in information-based industries. Products of the information economy differ from traditional manufactured products in the cost and time imposed on a reverse engineer. With manufactured goods, much of the know-how required to make manufactured goods remains within the factory when the products go to market, so that reverse engineering can only capture some of the product's know-how. Information-rich products of the digital economy, in contrast, bear a higher quantum of applied know-how in the product distributed in the market.¹⁵

¹³ 17 U.S.C. sec. 1201. There is a limited exception to enable bypassing technical controls and making tools to enable this when necessary to achieve interoperability among programs. See 17 U.S.C. sec. 1201(f). This law is discussed in Section V.

¹⁴ See, e.g., J.H. Reichman, *Legal Hybrids Between the Patent and Copyright Paradigms*, 94 Colum. L. Rev. 2432, 2438-40, 2506-11 (1994). Reichman has been a pioneer among intellectual property scholars in probing the tacit role of trade secret law in providing lead-time to innovators. Costliness itself will also suffice even without lead time, as discussed in Section II.

¹⁵ We build here on prior work distinguishing the accessibility of know-how in information-based industries as compared with traditional manufacturing industries. See, e.g., Reichman, *supra* note 1, at 660: (“[T]oday’s most productive and refined technical innovations are among the easiest of all forms of industrial know-how to duplicate. Because each product of the new technologies tends to bear its know-how on its face, like an artistic work, each is exposed to instant predation when successful and is likely to enjoy zero lead time after being launched on the market.”). See also Reichman, *Legal Hybrids*, *supra* note 14, at 2511-18; Pamela Samuelson, Randall Davis, Mitchell D. Kapor, & J.H. Reichman, *A Manifesto Concerning the Legal Protection of Computer Programs*, 94 Colum. L. Rev. 2308, 2314 (1994) (characterizing software as an information product that is more vulnerable than traditional manufactured goods to market-destructive appropriations because of the applied industrial know-how borne on or near the surface of software products) (cited hereinafter as “Manifesto”). See also Rochelle Cooper Dreyfuss, *A Wiseguy’s Approach to Information Products: Muscling Copyright and Patent Into a Unitary Theory of Intellectual Property*, 1992 Sup. Ct. Rev. 195 (1993).

For so-called “digital content” (movies, sound recordings, and the like), the relevant knowledge is entirely on the surface of the product, at least in the absence of technical protections such as encryption. Technical protections create costs for reverse engineers. When computer programs are distributed in object code form, a difficult analytical process is required to ascertain information embedded in the program, but it is there for the taking if a reverse engineer is willing to spend the time to study it.¹⁶ For computer chips, the relevant knowledge is circuit design, which is not only embodied within the chip, but readily accessible using technologies discussed below.¹⁷ The challenge is to design legal rules that protect information-rich products against market-destructive cloning, while at the same time providing enough breathing room for reverse engineering to enable new entrants to compete and innovate in a competitively health way.

Section III focuses on the semiconductor chip industry. When competitive reverse engineering and copying of semiconductor chip designs became too easy and too rapid to enable innovators to recoup research and development (R&D) costs, Congress responded by enacting the Semiconductor Chip Protection Act (SCPA) to protect chip makers from market-destructive cloning while at the same time affirming a right to reverse engineer chips, although limiting it to some degree.¹⁸ SCPA allows reverse engineers to copy circuit design to study it as well as to reuse information learned thereby in a new chip, but it imposes a forward-engineering requirement that inevitably increases a second comer’s development time and increases its costs.¹⁹ In the context of the chip industry, we think this restriction on reverse engineering was economically sound.

Section IV focuses on the software industry. Reverse engineering is undertaken for different reasons in the software industry than in other industrial contexts. The most economically significant reason to reverse engineer software, as reflected in the caselaw, is to learn information necessary to make a compatible program. The legal controversy over whether copies made of a program during the decompilation process infringed copyright was resolved in favor of reverse engineers. As Section IV explains, the economics of interoperability are more complex than legal commentators have acknowledged. However, on balance, we think that a legal rule in favor of reverse engineering of computer programs for purposes of interoperability is economically sound.

Section V discusses the emerging market for technically protected digital content. Because technical protection measures may be defeated by countermeasures, copyright industry groups persuaded Congress to enact the Digital Millennium Copyright Act that creates new legal rules reinforcing technical measures used by copyright owners to protect their works.²⁰ It protects them against most acts of circumvention, against the

¹⁶ See *infra* notes 181-83 and accompanying text.

¹⁷ See *infra* note 139 and accompanying text.

¹⁸ Pub. L. No. 98-620, 98 Stat. 3347 (1984), now codified at 17 U.S.C. sec. 901 et seq.

¹⁹ See 17 U.S.C. sec. 906(a), discussed *infra* notes 94-96 and accompanying text.

²⁰ Pub. L. No. 105-304, 112 Stat. 2860 (1998).

making and distribution of circumvention technologies, and against dissemination of information resulting from privileged acts of circumvention.²¹ In our view, these new rules restrict reverse engineering more than is economically sound, although the core idea of regulating trafficking in circumvention technologies may be justifiable.

Section VI steps back from particular industrial contexts and considers reverse engineering as one of the important policy levers of intellectual property law, along with rules governing term and scope of protection. The most obvious settings for the reverse engineering policy lever are “on” (reverse engineering is permissible) or “off” (reverse engineering is impermissible). However, our study reveals five additional strategies for regulating reverse engineering in the four industrial contexts studied: 1) regulating a particular means of reverse engineering; 2) adopting a “breadth” requirement for subsequent products; 3) permitting reverse engineering for some purposes but not others; 4) regulating tools for reverse engineering; and 5) restricting dissemination of information discerned from reverse engineering. We distinguish in this discussion between regulations affecting the act of reverse engineering and those affecting what the reverse engineer can do with the resulting information. Some restrictions on reverse engineering and on post-reverse engineering activities may be economically sound, although we caution against overuse of restrictions on reverse engineering because such restrictions implicate competition and innovation in important ways. Section VI also considers policy responses when innovators seek to thwart reverse engineering rights by contract or by technical obfuscation.

Intellectual property law in the United States has a deeply economic purpose of creating incentives to innovate as a means of advancing consumer welfare.²² The design of intellectual property rules, including those affecting reverse engineering, should be tailored to achieve these utilitarian goals, and should extend no farther than necessary to protect incentives to innovate. Intellectual property rights, if made too strong, might impede innovation and conflict with other economic and policy objectives.

II. Reverse Engineering in Traditional Manufacturing Industries

Reverse engineering is generally a lawful way to acquire know-how about manufactured products. Reverse engineering may be undertaken for many purposes.²³ We concentrate in this section on reverse engineering undertaken for the purposes of making a competing product because this is the most common and most economically significant reason to reverse engineer in this industrial context.²⁴ We argue that legal

²¹ The anticircumvention rules of the DMCA are now codified at 17 U.S.C. sec. 1201.

²² See, e.g., *Mazer v. Stein*, 347 U.S. 201, 219 (1954) (“The economic philosophy behind the clause empowering Congress to grant patents and copyrights is the conviction that encouragement of individual effort by personal gain is the best way to advance public welfare through the talents of authors and inventors.”)

²³ Pooley identifies six reasons for engaging in reverse engineering: 1) learning, 2) changing or repairing a product, 3) providing a related service, 4) developing a compatible product, 5) creating a clone of the product, and 6) improving the product. See Pooley, *supra* note 1, at 5-18 to 5-19.

²⁴ Reverse engineering undertaken for purposes of repairing a purchased product may well affect the manufacturer’s aftermarkets (e.g., for spare parts or service), but this will generally have less of an

rules favoring the reverse engineering of manufactured products have been economically sound because an innovator is nevertheless protected in two ways: by the costliness of reverse engineering and by lead-time due to difficulties of reverse engineering.²⁵ If technological advances transform reverse engineering so that it becomes a very cheap and rapid way to make a competing product, innovators may not be able to recoup their R&D expenses, and hence some regulation may be justified. An example discussed below is the plug-molding of boat hulls.

A. A Legal Perspective on Reverse Engineering

Reverse engineering has always been a lawful way to acquire a trade secret, as long as “acquisition of the known product...[is] by fair and honest means, such as purchase of the item on the open market.”²⁶ As the Restatement of Unfair Competition points out, “[t]he owner of a trade secret does not have an exclusive right to possession or use of the secret information. Protection is available only against a wrongful acquisition, use or disclosure of the trade secret,”²⁷ as when the use or disclosure breaches an implicit or explicit agreement between the parties or when improper means, such as trespass or deceit, are used to obtain the secret.²⁸ Even when a firm has misappropriated another firm’s trade secret, injunctive relief may be limited in duration based in part on the court’s estimation of how long it would take a reverse engineer to discover the secret lawfully.²⁹

The legal “right” to reverse engineer a trade secret is so well-established that courts nor commentators have rarely perceived a need to explain the rationale for this doctrine. A rare exception is the 1989 U.S. Supreme Court decision, *Bonito Boats, Inc. v. Thunder Craft Boats, Inc.*, which characterized reverse engineering as “an essential part of innovation,” likely to yield variations on the product that “could lead to significant advances in technology.”³⁰ Moreover, “the competitive reality of reverse engineering

economic effect on the manufacturer than if the reverse engineer makes a competing product. Reverse engineering to achieve compatibility will be discussed *infra* Section IV-B.

²⁵ This section focuses on incentives to invest in innovation in manufacturing industries when patent rights are not available (e.g., because the innovation is too modest an advance to meet the nonobviousness standard) or when firms choose trade secrecy to patents (e.g., because they do not want to disclose the innovation to the public as would be necessary to get a patent). Patents play an important role in creating incentives to invest in innovation, but innovators must recoup R&D expenses regardless of whether patent rights are available.

²⁶ Official Comment on Sec. 1 of Uniform Trade Secrets Act (cited hereinafter as UTSA).

²⁷ AMERICAN LAW INSTITUTE, RESTATEMENT OF THE LAW OF UNFAIR COMPETITION, comment a to Sec. 43 at 493 (1993) (cited hereinafter as “Restatement of Unfair Competition”). See, e.g., *Tabor v. Hoffman*, 118 N.Y. 30, 23 N.Y. 12 (Ct. App. 1889)(finding misappropriation of trade secrets where defendant exceeded authorized access to plaintiff secret patterns for making pumps to repair them by measuring and copying the patterns in order to make competing pumps).

²⁸ AMERICAN LAW INSTITUTE, RESTATEMENT OF TORTS, sec. 757 (1939); UTSA, *supra* note 26, sec. 1.

²⁹ See, e.g., Heald, *supra* note 7, at 975.

³⁰ 489 U.S. 141, 160 (1989), discussed *infra* Section II-C. See also MATTHEW JOSEPHSON, EDISON: A BIOGRAPHY 91 (1992) (“When the devices of others were brought before him for inspection, it was seldom that [Edison] could not contribute his own technical refinements or ideas for improved mechanical construction. As he worked constantly over such machines, certain original insights came to him; by dint

may act as a spur to the inventor” to develop patentable ideas.³¹ Even when reverse engineering does not lead to additional innovation, the *Bonito Boats* decision suggests it may still promote consumer welfare by providing consumers with a competing product at a lower price.³²

Further justification for the law’s recognition of a right to reverse engineer likely derives from purchase of the product in the open market which confers on its owner personal property rights, including the right to take the purchased product apart, measure it, subject it to testing, and the like. The time, money, and energy that reverse engineers invest in analyzing products may also be a way of “earning” rights to the information they learn thereby. Still another justification stems from treating the sale of a product in the open market as a kind of publication of innovations it embodies. This publication dedicates these innovations to the public domain unless the creator has obtained patent protection for them.³³

Courts have also treated reverse engineering as an important factor in maintaining balance in intellectual property law. Federal patent law allows innovators to have up to twenty years of exclusive rights to make, use and sell the invention,³⁴ but only in exchange for disclosure of significant details about their inventions to the public.³⁵ This deal is attractive in part because if an innovator chooses to protect its invention as a trade secret, such protection may be short-lived if it can be reverse-engineered. If state legislatures tried to make trade secrets immune from reverse engineering, this would undermine federal patent policy because it would “convert the [] trade secret into a state-conferred monopoly akin to the absolute protection that a federal patent affords.”³⁶ Reverse engineering is, then, an important part of the balance implicit in trade secret law.

of many trials, materials long known to others, constructions long accepted, were ‘put together in a different way’---and there you had an *invention*.”) (emphasis in original).

³¹ *Bonito Boats*, 489 U.S. at 160.

³² See, e.g., Heald, *supra* note 7, at 970. The Supreme Court did not make this point as directly as Heald, although it emphasized the right of the public to make use of unpatented designs in general circulation. See *Bonito Boats*, 489 U.S. at 164-65.

³³ See, e.g., *Tabor v. Hoffman*, 118 N.Y. 30, 23 N.E. 12 (Ct. App. 1889)(discussing the “publication” theory).

³⁴ 35 U.S.C. sec. 271(a).

³⁵ 35 U.S.C. sec. 112 (setting forth disclosure requirements). The Supreme Court in *Kewanee* spoke of patent law’s disclosure requirement as “the quid pro quo of the right to exclude.” *Kewanee*, 416 U.S. at 484. See also *id.* at 484-92 (emphasizing the importance of disclosure in achieving federal patent objectives and weaknesses in trade secrecy law, including the right to reverse engineer, as reasons why trade secrecy law does not conflict with federal patent policy).

³⁶ See *Chicago Lock Co. v. Fanberg*, 676 F.2d 400, 404 (9th Cir. 1981). *Fanberg* relies on the Supreme Court’s decision in *Kewanee*, 416 U.S. 470, in support of this position. *Kewanee* considered whether state trade secrecy law was in conflict with federal patent policy such that it should be preempted by this federal law. The majority in *Kewanee* concluded that no serious conflict existed because trade secrecy law was both weaker and different from patent law. Reverse engineering was one of the features of trade secrecy law that made it weaker and different from patent law. See *Kewanee*, 416 U.S. at 489-90. See also Pooley, *supra* note 1, at 5-16 (because reverse engineering makes trade secret law weaker than patent law, trade secret law is not preempted by patent law); 1 JAGER ON TRADE SECRETS sec. 5.04[3] at 5-39 (“The likelihood that unpatented objects in the public domain will be reverse engineered is part of the federal balance. It is an inducement to create patentable inventions.”). See also *Rockwell Graphic Systems, Inc. v.*

No reverse engineering right, as such, exists in patent law.³⁷ In theory, there should be no need to reverse engineer a patented invention to get information about how to make it because the patent specification should inform the relevant technical community of how to make the invention, and indeed the best mode of making it.³⁸ Insofar as a patent does not teach technologists everything they might want to know, it is clear that some reverse engineering activities will not infringe a patent. The purchaser of a machine embodying a patented invention, for example, is generally free to disassemble it to study how it works under the first sale principle of patent law.³⁹ In addition, a person who tries to make a patented invention to satisfy scientific curiosity may assert an experimental use defense to patent infringement.⁴⁰

Until quite recently, copyright law neither had nor had need for a reverse engineering privilege. The artistic and literary works this law traditionally protected did

DEV Industries, Inc., 925 F.2d 174, 178-80 (7th Cir. 1991)(discussing reverse engineering as a limitation on trade secret protection).

³⁷ See, e.g., Cohen & Lemley, *supra* note 12, at 6. Although there is no reverse engineering right, as such, in another U.S. intellectual property rights law, the Plant Variety Protection Act, 7 U.S.C. sec. 2321 et seq., there is a research exemption that serves a similar function: “The use and reproduction of a protected variety for plant breeding or other bona fide research shall not constitute an infringement of the protection provided under this Act.” *Id.* at sec. 2544. “The Research Exemption allows anyone to use the PVPA protected lines in a laboratory or field breeding research program to develop new lines. For example, a second comer may purchase a commercially available, PVPA protected soybean variety and use it to develop a new line. This new line can be sold or applied [sic] for protection of its own as long as it is new, distinct, uniform, and stable.” Memorandum of Christine Duh, pp. 2-3, Aug. 6, 2001 (on file with the authors).

³⁸ 35 U.S.C. sec. 112.

³⁹ See, e.g., Cohen & Lemley, *supra* note 12, at 30-35. By purchasing a manufactured product, the owner acquires the right to use it. Since disassembling a manufactured product does not involve making or selling the invention, no patent rights are implicated by reverse engineering in this context. See *infra* note 173-74 for a discussion of special characteristics of computer software that suggest that disassembly of this kind of product may implicate patent rights.

While disassembly of a manufactured product is generally lawful, some courts have sometimes enforced a contractual restriction on reverse engineering. See *K&G Oil & Tool Service Co. v. G&G Fishing Tool Service*, 158 Tex. 94, 314 S.W.2d 782, 785-86 (1958)(enforcing a negotiated agreement not to disassemble K&G’s magnetic fishing tool against competitor who then developed substantially the same tool). See also *Pioneer Hi-Bred Int’l, Inc. v. DeKalb Genetics Corp.*, 51 U.S.P.Q. 1987 (S.D. Iowa 1999)(enforcing “bag tag” prohibiting purchasers of PVPA-protected corn seed from using the seed for breeding or research purposes). For further discussion of the enforceability of contractual restrictions on reverse engineering in the computer software industry context, see *infra* Section IV-D.

⁴⁰ See, e.g., ROGER M. MILGRIM, *MILGRIM ON TRADE SECRETS* sec. 1.05[3] at 1-250 (2000). In U.S. patent law, the experimental use defense is quite narrow, not encompassing, for example, scientific or research use that may lead to development of a patentable invention or a commercial product. See, e.g., Rebecca S. Eisenberg, *Patents and the Progress of Science: Exclusive Rights and Experimental Use*, 56 U. Chi. L. Rev. 1017 (1989)(arguing for a broader experimental use defense in U.S. patent law). Exempting experimental uses of inventions from the scope of the patent right has achieved considerable acceptance in the international community. See, e.g., Janice M. Mueller, *No “Dilettante Affair”*: *Rethinking the Experimental Use Exception to Patent Infringement for Biomedical Research Tools*, 76 Wash. L. Rev. 1, 37-39 (2001).

not need to be reverse-engineered to be understood.⁴¹ Books, paintings, and the like bear the know-how they contain on the face of the commercial product sold in the marketplace. To access this information, one simply needed to read or analyze the work. Moreover, at least till the admission of computer programs to its domain, copyright law did not protect industrial products of the sort that firms typically reverse-engineer.⁴²

B. An Economic Perspective on Reverse Engineering

The economic effects of reverse engineering depend on a number of factors, including the purpose for which it is undertaken, the industrial context within which it occurs, how much it costs, how long it takes, whether licensing is a viable alternative, and how the reverse engineer uses information learned in the reverse engineering process.⁴³ In this subsection, we concentrate on the economics of reverse engineering undertaken for the purpose of developing a competing product.⁴⁴

We argue that a legal right to reverse engineer does not typically threaten an innovative manufacturer because it generally has two forms of protection against a reverse-engineering competitors: lead-time before reverse engineers can enter⁴⁵ and costliness of reverse engineering. Lead-time serves the same function as a short-lived intellectual property right. Costliness may prevent reverse engineering entirely, especially if the innovator licenses others as a strategy for preventing unlicensed entry. Provided that the cost of reverse engineering is high enough, such licensing will be on terms that permit the innovator to recoup its R&D expenses, while at the same time constraining the exercise of market power in order to dissuade other potential entrants.

Our economic assessment of reverse engineering recognizes that this activity is only one step in what is typically a four-stage development process. The first stage of a

⁴¹ See Section IV-A for a discussion of the controversy in copyright law over the legality of reverse engineering of computer software, a non-traditional copyright subject matter that does not reveal its know-how on the face of mass-market products.

⁴² Pictorial, sculptural or graphic works can be protected by U.S. copyright law unless they have a utility beyond conveying information or displaying of an appearance. See 17 U.S.C. sec. 101 (definitions of pictorial, sculptural and graphic works and of useful article). Many industrial products (e.g., chairs, automobiles and toasters) have an aesthetic appearance, yet not be copyrightable in the U.S. because their aesthetic design is not separable from the utilitarian functions the products have. See, e.g., *Brandir Int'l, Inc. v. Cascade Pacific Lumber Co.*, 834 F.2d 1142 (2d Cir. 1987) (aesthetic design for bicycle rack uncopyrightable because of inseparability of functional considerations in final design).

⁴³ Reverse engineering does not of itself render the trade secret valueless because reverse engineers do not generally publish their discoveries, but instead maintain the discovered information as their own trade secret. See, e.g., Pooley, *supra* note 1, at 5-19. If reverse engineers do publish the information, this can erode an innovator's ability to recoup its R&D expenses because the secret will have gotten out.

⁴⁴ Some economic effects arising from reverse engineering for purposes of developing complementary products are explored *infra* Section IV-B.

⁴⁵ Empirical studies of manufacturing firms over a long period demonstrate that such firms rely more on lead-time than on patents as the principal source of protection for their intellectual assets. See, e.g., Wesley M. Cohen, Richard R. Nelson, & John P. Walsh, *Protecting Their Intellectual Assets: Appropriability Conditions and Why U.S. Manufacturing Firms Patent (or Not)* (Feb. 2001)(manuscript on file with the author). See also Reichman, *Legal Hybrids*, *supra* note 12, at 2439-41 (explaining the importance of lead-time in trade secrecy law).

second comer's development process is an awareness stage.⁴⁶ This involves a firm's recognition that another firm has introduced a product into the market that is potentially worth the time, expense and effort of reverse engineering. In some markets, recognition happens very rapidly; in others, it may take some time, during which the innovator can begin to recoup its R&D costs by selling its product and establishing good will with its customer base.⁴⁷

Second is the reverse engineering stage. This begins when a second comer obtains the innovator's product and starts to disassemble and analyze it to discern of what and how it was made.⁴⁸ The reverse engineering stage may be costly, time-consuming, and difficult,⁴⁹ although this varies considerably, depending mainly on how readily the innovator's product will yield the know-how required to make it when confronted by a determined and skilled reverse engineer.⁵⁰ However, a reverse engineer will generally

⁴⁶ The more innovative the product, the longer it may take for potential competitors to recognize the innovation and undertake to copy it. However, the innovator may also find it difficult to achieve initial market success. See, e.g., GEOFFREY A. MOORE & REGIS MCKENNA, *CROSSING THE CHASM: MARKETING AND SELLING HIGH TECHNOLOGY PRODUCTS TO MAINSTREAM CUSTOMERS* (1991). Because of this, the more innovative the product, the more economically sensible it will generally be to obtain patent protection for key aspects of the innovation to impede competitive imitation.

⁴⁷ For some consumers, a firm's reputation for innovation or quality service will make its product attractive even if second comers eventually copy it. To the extent there are switching costs associated with the product (e.g., owing to a steep learning curve in how to use it), the innovator may also benefit from "lock-in" of its initial customers and those who later value the innovator's product because others are using it. See, e.g., Mark A. Lemley & David McGowan, *The Law and Economics of Network Effects*, 86 Calif. L. Rev. 479 (1998).

⁴⁸ The reverse engineer's purchase of a competitor's product to reverse engineer it does, of course, make some contribution toward recoupment of the innovator's costs; this may be trivial, however, in the case of many mass-market goods.

⁴⁹ Products vary considerably in the ease with which they can be reverse engineered. In general, the more difficult reverse engineering is, the greater value the secret will have, the longer lead-time advantage the trade secret holder will enjoy in the market, and the less incentive the holder may have to license the secret. See, e.g., Pooley, *supra* note 1, at 4-42. See also Restatement, *supra* note 27, sec. 39, com. f. Firms can sometimes make reverse engineering more difficult, and this may be an economically sensible thing to do if the secret is valuable. See, e.g., Pooley, *supra* note 1 at 5-25: "It may be possible to build products that are difficult to break down and copy. Hardware components can be encapsulated to make nondestructive disassembly almost impossible; components can be mislabeled...; custom parts can be used; 'locks' (often implemented in software) can be added. In any sort of complex product, nonfunctional features can be added to create a 'fingerprint' on any illegitimate copy, forcing copyists to invest in real reverse engineering efforts." Friedman, Landes & Posner regard the expenditures required to make the product more difficult to reverse engineer as costs of not prohibiting reverse engineering. Friedman et al., *supra* note 2, at 70. Professor Kitch discusses other reasons it is difficult to "steal" valuable information. See Edmund W. Kitch, *The Law and Economics of Rights in Valuable Information*, 9 J. Legal Stud. 683, 711-15 (1980). See also Steven N.S. Cheung, *Property Rights in Trade Secrets*, 20 Econ. Inquiry 40, 47 (1982)(discussing economics of trade secrecy law and various means by which trade secret rents may be dissipated).

⁵⁰ See, e.g., Pooley, *supra* note 1, at 4-41. The relative difficulty of reverse engineering does not, of course, match up perfectly with the difficulty and expense of developing the secret in the first place. Some trade secrets may have been serendipitously developed at low cost, yet are difficult to reverse engineer, while other expensive and time consuming innovations may be impossible to hide in the final product. Still, some commentators contend that "inventiveness often correlates with difficulty of reverse engineering, with the result that the more inventive the product, the longer its inventor enjoys the so-called 'first mover advantage,' and the more profit she earns." ROCHELLE C. DREYFUSS & ROBERTA R. KWALL,

spend less time and money to discern this know-how than the initial innovator spent in developing it, in part because the reverse engineer is able to avoid wasteful expenditures investigating approaches that don't work, and in part because advances in technology typically reduce the costs of rediscovery over time.⁵¹

Third is the implementation stage.⁵² After reverse engineering the innovator's product, a second comer must take the know-how obtained during the reverse engineering process and put it to work in designing and developing a product to compete in the same market. This may involve making prototypes, experimenting with them, retooling manufacturing facilities, and reiteration of the design and development process until it yields a satisfactory product. It may be necessary to return to the reverse engineering stage again if it becomes apparent in the implementation phase that some necessary know-how eluded the reverse engineer the first time. Information obtained during reverse engineering may, moreover, suggest possibilities for additional product innovation that will be investigated in the implementation stage.⁵³ For these reasons, the second comer's implementation stage may take considerable time and require significant expense.

The fourth stage in the second comer's development process is the introduction of its product to the market. How quickly the new product will erode the innovator's market share and force the innovator to reduce prices to be competitive with the new entrant will depend on various market factors.⁵⁴

In the chart and discussion below, we use four criteria to assess the social welfare effects of the law's recognition of a right to reverse engineer. The criteria are the effects on: 1) incentives to innovate, 2) incentives to engage in follow-on innovation, 3) prices,

INTELLECTUAL PROPERTY: CASES AND MATERIALS ON PATENTS, COPYRIGHTS, AND TRADEMARKS 818 (1996).

A further consideration is how difficult or easy it is to detect whether another firm independently developed the same or a similar innovation or engaged in reverse engineering to discover it. Reverse engineering, after all, tends to occur behind closed doors. See Friedman et al., *supra* note 2, at 70; Kitch, *supra* note 49, at 690. However, it is sometimes possible to persuade courts that independent invention of the same trade secret was unlikely. See, e.g., *Pioneer Hi-Bred Int'l v. Holden Foundation Seeds*, 35 F.3d 1226 (8th Cir. 1994).

⁵¹ See, e.g., Friedman et al., *supra* note 2 at 63. See also JARED DIAMOND, GUNS, GERMS, AND STEEL (1999)(giving examples of technologies, the reinvention of which occurred rapidly once it became known that the technology was possible).

⁵² During both the reverse engineering and the implementation stages, the innovator may decide to license its know-how to the second comer. Over time, the innovator's willingness to license may increase, especially if it has reason to think that certain second comers are making progress toward developing a competing or improved product. The second comer's willingness to take a license may decline as his expenditures in reverse engineering and redevelopment rise and as it perceives these efforts to be bearing fruit. Yet, a license from the innovator may become attractive if fine details of implementation elude the reverse engineer.

⁵³ See, e.g., Richard C. Levin et al., *Appropriating Returns from Industrial Research and Development*, 1987 BROOKINGS PAPERS ON ECONOMIC ACTIVITY 783, 805-07 (improvements likely to result from reverse engineering).

⁵⁴ It bears repeating that an innovator may be able to hold on to its leading market share if it has a positive reputation for quality or service, it has a strong brand, or there are high switching costs.

and 4) socially wasteful expenditures of resources. At first glance, these considerations seem to cut in opposite directions in the manufacturing industry context. On the negative side, the right to reverse engineer seems to decrease incentives for first comers to introduce new products, and to encourage wasteful expenditures on reverse engineering.⁵⁵ On the positive side, a right to reverse engineer can increase competition in the marketplace, leading to lower prices, and can spur follow-on innovations by second comers.

However, the argument against reverse engineering that is based on wasted costs is misleading. The cost of reverse engineering can be avoided by licensing. Under the threat of reverse engineering, an innovator knows that its market position can be eroded by unlicensed entry. The firm can preempt this outcome by authorizing entry in a controlled way through licenses. Licensing should be in the interest of both the innovator and potential reverse engineers. Licensing allows innovators to recoup costs by charging licensing royalties or other fees, and in addition, the royalties preclude licensees from pricing their products in a market-destructive way.⁵⁶ Licensing can achieve the same knowledge-sharing and market outcomes as reverse engineering without incurring the costs of reverse engineering. The cost savings can be shared through the terms of the license.⁵⁷

Although the right to reverse engineer may *reduce* incentives to innovate, the important question is whether the incentive is *sufficient* when reverse engineering is permitted. The answer to this question will chiefly depend on the costs of reverse engineering relative to the innovator's development costs and on how long the process of reverse engineering takes. As pointed out above, costs of reverse engineering are often substantial and take considerable time.⁵⁸ Of course, if the costs of reverse engineering

⁵⁵ See, e.g., Martin J. Adelman, *Property Rights Theory and Patent-Antitrust: The Role of Compulsory Licensing*, 52 N.Y.U. L. Rev. 977, 982 (1977)(expressing concern about wasteful expenditures of reinvention). Another set of socially wasteful costs that may be incurred if reverse engineering is legal are the costs of making one's product difficult to reverse engineer. See supra note 43.

⁵⁶ See, e.g., Reichman, *Legal Hybrids*, supra note 14, at 2441. See also J.H. Reichman, *Of Green Tulips and Legal Kudzu: Repackaging Rights in Subpatentable Innovation*, 53 Vand. L. Rev. 1743 (2000) (proposing a compensatory system to enable developers of subpatentable innovations to recoup R&D expenses).

⁵⁷ This argument parallels an argument in Stephen Maurer & Suzanne Scotchmer, *The Independent Invention Defense in Intellectual Property*, John M. Olin Working Paper 98-11, U.C. Berkeley (Boalt Hall) (1998), forthcoming *Economica* (2002). These authors consider the consequences of allowing entry by independent inventors in markets with patented products. They argue that the threat to a rightholder's market depends on the cost of entry by rivals, in particular the cost of independent invention or inventing around a patent. Reverse engineering is just another costly way to enter the market. Reverse engineering differs from independent invention or inventing around a patent in that the product is typically not patented, and reverse engineering may be less costly than inventing around. Nevertheless, the effect of entry depends only on cost, and the same argument applies in all three contexts. That argument, in which the cost of entry is avoided by licensing, is repeated below for reverse engineering. The argument differs from previous arguments, e.g., Adelman supra note 55, in that unlicensed entry is assumed not to occur. Instead, the threat of entry affects the terms of license, which will be used by the rightholder for two purposes: to collect profit from authorized entrants, and to control the price of the product. The price will be just low enough to deter further (unauthorized) entry, but not lower.

⁵⁸ See supra notes 46-50 and accompanying text. Dreyfuss & Kwall put the point succinctly:

are very low in comparison with the costs of the initial development or if second comers can enter quickly because reverse engineering is easy, the second comer's entry will rapidly drive prices down due to the rival's cost advantage. This would likely deprive the innovator of revenues sufficient to cover its development costs.⁵⁹ The effect of a reverse engineering right on incentives to innovate will thus depend on the relative costs incurred by the innovator and potential reverse engineers and on the natural lead-time protection afforded by delay, not on whether reverse engineering is avoided by licensing.

Table 1 illustrates the social welfare effects of two possible reverse engineering rules in the context of traditional manufacturing industries: one allowing it and one disallowing it. As to each criterion, the effects of permitting reverse engineering are compared with the effects of forbidding it.

Table 1
Social Calculus of Reverse Engineering in Manufacturing Sector

	RE legal	RE illegal
Incentives to innovate	lower (but adequate)	higher (but excessive)
Price	lower	higher
Follow-on innovation	higher	lower

Because reverse engineering generally takes time (time to decide the product is worth figuring out as well as time to actually do the engineering and bring the product to market), the first inventor enjoys a period of exclusivity in which to recapture the costs of the invention, build a reputation, and establish a base of loyal customers. Furthermore, the copyist is not quite a free rider because reverse engineering is generally expensive. Thus, after the secret is discovered, the parties compete on a fairly level playing field.

Dreyfuss & Kwall, *supra* note 50, at 818.

⁵⁹ See, e.g., Wendy J. Gordon, *Asymmetric Market Failure and Prisoners' Dilemma in Intellectual Property*, 17 U. Dayton L. Rev. 853 (1992) (discussing conditions under which market failure may arise from appropriation of intellectual creations); Wendy J. Gordon, *On Owning Information: Intellectual Property and the Restitutory Impulse*, 78 Va. L. Rev. 149 (1992) (discussing the concept of "malcompetitive" copying).

Douglas Lichtman has argued that incentives to develop subpatentable innovations such as boat hulls will be threatened if there is a right to engage in very low cost reverse engineering, as by use of plug molds. See Douglas Lichtman, *The Economics of Innovation: Protecting Unpatentable Goods*, 81 Minn. L. Rev. 693, 721-23 (1997). Maurer and Scotchmer, *supra* note 57, argue from the other direction: incentives to innovate will survive a rival's independent innovation whenever its costs are roughly "commensurate" with the innovator's development cost.

One reason that the cost of reverse engineering can be very cheap relative to the innovator's cost is that the reverse engineer avoids "dry holes." This is particularly important in some industries. By some counts, only one in five attempts to develop a drug succeeds. The reverse engineer can work on those known to be viable and avoid the others. Fortunately, drugs are protected by patents, and are hence immune to market-destructive reverse engineering and reimplementation. Where that has not been true, as in India prior to the TRIPS Agreement, drugs were very cheap due to the ease of reverse engineering their chemical structure. See, e.g., Jean O. Lanjouw, *The Introduction of Pharmaceutical Product Patents in India: 'Heartless Exploitation of the Poor and Suffering?'*, NBER Working Paper no. 6366 (1998).

Four years later the U.S. Supreme Court overruled *Interpart* in *Bonito Boats, Inc. v. Thunder Craft Boats, Inc.*⁶⁷ One reason the Court gave for striking down Florida’s anti-plug mold law was that it “prohibits the entire public from engaging in a form of reverse engineering of a product in the public domain.”⁶⁸ The Court said that it was “difficult to conceive of a more effective method of creating substantial property rights in an intellectual creation than to eliminate the most efficient method for its exploitation.”⁶⁹ Drawing upon earlier preemption rulings, the Court said they protected “more than the right of the public to contemplate the abstract beauty of an otherwise unprotected intellectual creation—they assure its efficient reduction to practice and sale in the marketplace.”⁷⁰ It went on to say that “[w]here an item in general circulation is unprotected by a patent, ‘[r]eproduction of a functional attribute is legitimate competitive activity.’”⁷¹

The economic consequences of plug-molding deserved more serious consideration.⁷² The plug-mold process dramatically reduces the costs of and time required to engage in reverse engineering and reimplementing of an innovation. If plug-molding undermines incentives to invest in innovative boat hulls or other manufactured goods,⁷³ a ban on the use of the plug-mold process might be economically

⁶⁷ 489 U.S. 141 (1989).

⁶⁸ *Id.* at 160. *Bonito Boats* seems to elevate the principle of reverse engineering to a constitutionally protected interest. See, e.g., *Chicago Lock Co. v. Fanberg*, 676 F.2d 400, 404 (9th Cir. 1982) (for state law not to allow reverse engineering “would, in effect, convert the Company’s trade secret into a state-conferred monopoly akin to the absolute protection that a federal patent affords. Such an extension of California trade secrets law would certainly be preempted by the federal scheme of patent regulation.”) See also Reichman, *Legal Hybrids*, *supra* note 14, at 2473 (interpreting *Bonito Boats* as “endow[ing] the competitor’s right to reverse engineer with constitutional underpinnings”).

⁶⁹ *Bonito Boats*, 489 U.S. at 164.

⁷⁰ *Id.* The cases upon which the Court principally drew were the companion cases of *Sears, Roebuck & Co. v. Stiffel Co.*, 376 U.S. 225 (1964) and *Compco Co. v. Day-Brite Lighting, Inc.*, 376 U.S. 234 (1964). In these cases the Court ruled that state unfair competition law could not be used to protect unpatentable designs from competitive copying because this would interfere with federal patent policy. Although the courts have been consistently hostile to unfair competition-like claims as a means to protect unpatented designs since *Sears* and *Compco*, they have been far more receptive to protecting product configurations against copying under trade dress law. See, e.g., *Sunbeam Products, Inc. v. West Bend Co.*, 123 F.3d 246 (5th Cir. 1997). The Supreme Court has endorsed trade dress claims for product configurations or designs in appropriate cases; yet it has placed a heavy burden of proof on a trade dress claimant to show that the claimed configuration or design is nonfunctional if it was claimed in an expired patent. See *TrafFix Devices, Inc. v. Marketing Displays, Inc.*, 532 U.S. 23 (2001).

⁷¹ *Bonito Boats*, 489 U.S. at 164. It should be noted that in 1998 Congress enacted a new form of intellectual property protection for vessel hulls. See 17 U.S.C. sec. 1301 et seq. Now they can neither be plug-molded nor copied by any other method.

⁷² Like the Court, economists would be concerned about the distortions likely to arise from non-uniform state laws. As the framers of the U.S. Constitution understood very well, states are not well-equipped to provide effectual protection of publicly disclosed innovations. It is for this reason that the framers included Art. I, sec. 8, cl.8, in the U.S. Constitution. See THE FEDERALIST No. 43, at 338 (James Madison) (John C. Hamilton, ed., 1804) (1788). The non-uniformity problem was present in the *Bonito Boats* case because Thunder Craft Boats was a Tennessee-based company and Tennessee had no anti-plug molding statute. See *Bonito Boats*, 489 U.S. at 145.

⁷³ It should not be enough for boat designers to testify that they needed such a law. Robert Kastenmeier, former head of the Intellectual Property Subcommittee of the House Judiciary Committee, recognized the danger of new laws to protect particular industries. It is very easy for special interest groups to claim that

sound, at least for some period of time. The germ of an argument that plug-molding might have market-destructive effects can be found in *Bonito Boats*. The Supreme Court noted that Bonito Boats had expended substantial resources in developing the boat hull that it sought to protect in the litigation against Thunder Craft Boats,⁷⁴ and that the very purpose of the plug mold process was to “provide a method for making large molded boat hull molds at very low cost, once a prototype hull has been provided.”⁷⁵ Yet the Court gave very little attention to these details in its lengthy legal and policy analysis of the case.

The Supreme Court suggested in *Bonito Boats* that plug mold duplication of boat hulls was, “an essential part of innovation in the field of hydrodynamic design.”⁷⁶ Professor Heald has questioned this assertion, pointing out that the Florida law “primarily discriminates against those interested in reproduction not innovation”⁷⁷ and that plug-molding might well “result in less innovation.”⁷⁸ Heald’s is the more economically sound view of the effects of plug-molding on follow-on innovation.⁷⁹

Of course, this does not mean that the laws enacted in Florida or California were adopted on the basis of economic merit. Some features of the Florida law suggest that it was the product of a rent-seeking special interest group lobby. Consider, for instance, that the law applied retroactively to boat hulls already in existence.⁸⁰ Moreover, it did

they need more legal protection, but this does not mean that adopting such a law is necessarily in the overall public interest. To guard against special interest lobbying, Kastenmeier articulated a multi-part test to determine when legislation of this sort would be warranted. See Robert W. Kastenmeier & Michael J. Remington, *The Semiconductor Chip Protection Act of 1984: Swamp or Firm Ground?*, 70 Minn. L. Rev. 417, 438-61 (1985).

⁷⁴ *Bonito Boats*, 489 U.S. at 144.

⁷⁵ *Id.* at 164 (quoting from the patent).

⁷⁶ *Id.* at 160.

⁷⁷ Heald, *supra* note 7, at 987. See also Reichman, *Legal Hybrids*, *supra* note 14, at 2473 (plug-molders merely duplicate originator’s product).

⁷⁸ Heald, *supra* note 7, at 987. The Court insisted that enactment of laws to give incentives to invest in innovation is reserved to the federal government, not to states. *Bonito Boats*, 489 U.S. at 157-58. Heald reinforces the Court’s position by asserting that “the Constitution grants Congress the right to experiment in the area. Congress’ intent is frustrated by state statutes whose incentives interfere with Congress’ experiments.” Heald, *supra* note 7, at 969. However, many state laws, including those that protect trade secrets, trademarks, and rights of publicity, aim, in part, at inducing investment in intellectual creations; yet, they are generally not preempted. See, e.g., John S. Wiley, Jr., *Bonito Boats: Uninformed But Mandatory Innovation Policy*, 1989 Sup. Ct. Rev. 283, 290-94 (1989)(discussing state intellectual property laws threatened by preemption analysis in *Bonito Boats*).

⁷⁹ If reverse engineering is a process that results in discovery of know-how, not just rapid, cheap copying of existing products, one might argue that plug-molding is not reverse engineering at all. As subsection A has shown, reverse engineering and competitive copying of a product are different activities, even if courts, as in *Bonito Boats*, sometimes conflate them. See, e.g., *Bonito Boats*, 489 U.S. at 160 (Florida law “prohibits the entire public from engaging in a form of reverse engineering of a product in the public domain”); *Traffix Devices, Inc. v. Marketing Displays, Inc.*, 532 U.S. 23 (2001)(seeming to conflate reverse engineering and copying). By pointing out this difference, we do not mean to suggest that cloning is always or necessarily economically harmful. As long as the costs of cloning are roughly commensurate with the costs of initial development or there is enough delay in the cloner’s entry so that the first comer can recoup R&D costs, introduction of an identical product can be economically beneficial.

⁸⁰ Retroactive application of the law cannot incent the creation of existing designs. It is worth pointing out that *Bonito Boats* developed the 5VBR boat more than six years before the Florida legislature passed the

not require any showing of originality, novelty, or improvement as a criterion for the grant of protection.⁸¹ Nor was there any durational limit to the protection.⁸² It is difficult to believe that perpetual rights are necessary to enable boat hull designers to recoup their R&D expenses.⁸³ An economically sound anti-plug mold law might, then, apply only prospectively, have a minimal creativity requirement, and a durational limitation aimed at providing a reasonable amount of lead-time to enable innovators to recoup its investments, but not more than that.⁸⁴ In 1998, Congress enacted a “sui generis” (of its own kind) form of intellectual property protection to protect boat hulls from unauthorized copying, not just from plug-molding.⁸⁵

From an economic perspective, anti-plug mold laws illustrate that even in the context of traditional manufacturing industries, a form of reverse engineering and reimplementaion that produces cheap, rapid identical copies has the potential to have market-destructive consequences. “Quick imitation robs innovation of value.”⁸⁶ Insofar as market-destructive effects can be demonstrated, it may be economically sound for the law to restrict a market-destructive means of reverse engineering and reimplementaion for a period of time sufficient to enable the innovator to recoup its R&D expenses. Plug-molding is only one example of technological advances that have changed the economic calculus of reverse engineering rules, as subsequent sections will show.

III. Reverse Engineering In the Semiconductor Industry

anti-plug mold law, yet the law protected this hull as well as all new designs. *Bonito Boats*, 489 U.S. at 144-45.

⁸¹ *Id.* Heald was critical of the Florida plug mold law for the lack of a creativity requirement. See Heald, *supra* note 2, at 987.

⁸² *Bonito Boats*, 489 U.S. at 144-45. Some commentators have been critical of the Florida plug mold law on this basis as well. See, e.g., Lichtman, *supra* note 59, at 718.

⁸³ By the time Thunder Craft copied the 5VBR boat hull and sold competing boats, Bonito Boats had already had eight years within which to recoup its R&D expenses on that hull. *Bonito Boats*, 489 U.S. at 144-45.

⁸⁴ The new form of intellectual property right Congress enacted in 1998 to protect boat hulls does have an originality requirement and a durational limitation. See 17 U.S.C. secs. 1302 (originality requirement), 1305 (duration limitation).

⁸⁵ See Vessel Hull Design Protection Act, which was Title V of the Digital Millennium Copyright Act, Pub. L. No. 105-304, 112 Stat. 2860 (1998), now codified at 17 U.S.C. sec. 1301 et seq. In protecting the configuration of boat hulls, the VHDPA most closely resembles utility model laws adopted in some countries. See, e.g., Reichman, *Legal Hybrids*, *supra* note 14, at 2455-59 (discussing utility model laws). For the moment, the VHDPA only covers vessel hulls, but only minor changes would be necessary to convert it to a more general intellectual property law to protect the configuration of manufactured products. Congress has rejected legislation of this sort in nearly every session during the 20th century because of concerns it would unduly impede competition in product markets. See, e.g., Richard Frenkel, *Intellectual Property Law in the Balance: Proposals for Improving Industrial Design Protection in the Post-TRIPS Era*, 32 Loy. L.A. Law Rev. 531 (1999). For a discussion of industrial design protection more generally and why it has been controversial over the years, see, e.g., J.H. Reichman, *Design Protection and the New Technologies: The United States Experience in a Transnational Perspective*, 19 U. Balt. L. Rev. 6 (1989-90). However, the expansion of state and federal trade dress protection for product configurations has had much the same effect as an industrial design would have in the U.S. *Id.* The functionality limitation on trade dress protection limits the utility of this law as a surrogate for a European-style utility model law.

⁸⁶ Email communication from Michael Moradzadeh (former executive at Intel Corp.) to Pamela Samuelson (on file with the authors).

The semiconductor industry is in many respects a traditional manufacturing industry. However, we give it separate treatment here for two reasons. First, semiconductors are information technology products that bear a high quantum of the know-how required to make them on the face of the product sold in the market.⁸⁷ This made them vulnerable to rapid, cheap competitive cloning that industry leaders asserted undermined their ability to recoup the very high costs of R&D necessary to produce new chips.⁸⁸ Second, Congress responded to these industry concerns about “chip piracy”⁸⁹ by creating a new form of intellectual property protection for semiconductor chip designs.⁹⁰

The Semiconductor Chip Protection Act (SCPA)⁹¹ is noteworthy for a number of reasons.⁹² First, it is one of the few intellectual property laws⁹³ with an express reverse

⁸⁷ See, e.g., Reichman, *Legal Hybrids*, supra note 14, at 2479-80, Manifesto, supra note 15, at 2338 (discussing the vulnerability of information technology products to market-destructive appropriations because of the high quantum of know-how they bear on or near the face of products sold in the marketplace). See also Morton David Goldberg, *Semiconductor Chip Protection As a Case Study*, in GLOBAL DIMENSIONS OF INTELLECTUAL PROPERTY RIGHTS IN SCIENCE AND TECHNOLOGY (Mitchel B. Wallerstein, Mary Ellen Moguee, & Roberta A. Schoen, eds. 1993) at 333 (“Considerable skill and creativity are invested in the design of the mask works that determine the topography of those products, but this design work is easily appropriated since, in essence, each copy of the product carries its own blueprint with it.”)

⁸⁸ See, e.g., Prepared Testimony of F. Thomas Dunlap, Jr., Corporate Counsel and Secretary of Intel Corp. (cited hereinafter as “Dunlap Statement”), Hearings Before the Subcommittee on Courts, Civil Liberties, and the Administration of Justice of the House Committee on the Judiciary, on H.R. 1028, 98th Cong., 1st Sess. (8/3/83) (cited hereinafter as “House Hearings”) explaining the industry’s need for this legislation.

⁸⁹ See, e.g., Prepared Statement of Sen. Charles McC. Mathias, Jr., House Hearings, supra note 88, at 3: “[Chip] innovators are being ripped off by onshore and offshore ‘chip pirates’ who, for a fraction of the developer’s cost, can now legally appropriate and use these chip designs as their own.” Of particular concern was the loss to Japanese industry of a substantial share of the market for random access memory chips to Japanese competitors whose superior quality control made their chips very competitive. See Stephen P. Kasch, *The Semiconductor Chip Protection Act: Past, Present, and Future*, 7 High Tech. L.J. 71, 79 (1993).

⁹⁰ Some commentators have suggested that the semiconductor industry “greatly overstated the severity of the chip piracy problem” in testimony before Congress. See, e.g., Robert Risberg, *Five Years Without Infringement Litigation Under the Semiconductor Chip Protection Act: Unmasking the Specter of Chip Piracy in an Era of Diverse and Incompatible Technologies*, 1990 Wisc. L. Rev. 241, 244-45 (1990). See also Kasch, supra note 89, at 92 (questioning evidence of chip piracy at legislative hearings).

⁹¹ Pub. L. No. 98-620, 98 Stat. 3347 (1984), now codified at 17 U.S.C. sec. 901 et seq.

⁹² SCPA has been the subject of much commentary. See, e.g., Kasch, supra note 88; John G. Rauch, *The Realities of Our Times: The Semiconductor Chip Protection Act of 1984 and the Evolution of the Semiconductor Industry*, 3 Fordham Ent., Media & Intell. Prop. L.F. 403 (1993); Risberg, supra note 85; Linda Samuels & Jeffrey M. Samuels, *Semiconductor Chip Protection Act of 1984: An Analytical Commentary*, 23 Am. Bus. L. J. 601 (1986); Terrill Lewis, *Comment, Semiconductor Chip Process Protection*, 32 *Hous. L. Rev.* 555 (1995). See also *Symposium: The Semiconductor Chip Protection Act of 1984 and Its Lessons*, 70 *Minn. L. Rev.* 263 (1985) (consisting of six articles); RICHARD H. STERN, SEMICONDUCTOR CHIP PROTECTION (1986); ANDREW CHRISTIE, INTEGRATED CIRCUITS AND THEIR CONTENTS: INTERNATIONAL PROTECTION (1995).

⁹³ Although trade secrecy is sometimes characterized as a form of intellectual property protection, see, e.g., Stanley M. Besen & Leo J. Raskind, *An Introduction to the Law and Economics of Intellectual Property*, 5 *J. Econ. Perspectives* 3, 3 (1991), it is more appropriately understood as a branch of unfair competition law. See, e.g., Restatement of Unfair Competition, supra note 27, secs. 39-44. Trade secret law confers no

engineering privilege.⁹⁴ Second, the privilege permits the copying of protected chip designs in order to study the layouts of circuits, and also incorporation of know-how discerned from reverse engineering in a new chip.⁹⁵ Third, the SCPA requires reverse engineers to engage in enough “forward-engineering” to develop an original chip design that itself qualifies for SCPA protection.⁹⁶ This is in contrast with the predominant legal rule for manufacturing industries that permits reverse engineers to make and sell identical or near-identical products to those they have reverse engineered.⁹⁷ The economic rationale for the forward engineering requirement was not articulated with precision during the SCPA debate, but we think it is fundamentally sound as applied to this industry.

A. Perturbations in Product Life Cycles in the Chip Industry

The typical product life cycle in the semiconductor industry was relatively constant in the 1970’s and 1980’s.⁹⁸ A pioneering firm, usually Intel Corp., would develop an innovative new product and introduce it to the market priced handsomely so that the firm could recoup its investments. “Later, as the manufacture [became] more efficient [the innovator would cut] its prices to expand its market and discourage competition. Nonetheless, second-source products—chips electrically and mechanically compatible with the pioneering product—eventually appear[ed] on the market. The arrival of competition precipitate[d] further rounds of price cuts.”⁹⁹ Toward the end of this life cycle, the pioneer’s profit margins would tail off, and it would have to hope that the next round of innovation would allow it to regain market share and profits.

exclusive rights on innovators, as intellectual property statutes typically do, but only protects holders from certain kinds of tortious acts, such as use of improper means or breach of confidence to acquire the secret.

⁹⁴ Professor Raskind has spoken of the reverse engineering privilege as the “capstone” of SCPA. See Leo J. Raskind, *Reverse Engineering, Unfair Competition, and Fair Use*, 70 Minn. L. Rev. 385, 385 (1985). The reverse engineering privilege of SCPA also received attention in other articles, including Harold R. Brown, *Fear and Loathing of the Paper Trail: Originality in Products of Reverse Engineering Under the Semiconductor Chip Protection Act As Analogized to the Fair Use of Nonfiction Literary Works*, 41 Syr. L. Rev. 985 (1990) and Lee Hsu, *Reverse Engineering Under the Semiconductor Chip Protection Act: Complications for Standard of Infringement*, 5 Albany L.J. Sci. & Tech. 249 (1996).

⁹⁵ 17 U.S.C. sec. 906(a). Indeed, a Congressional explanatory memorandum about SCPA states that chip designs produced by this sort of reverse engineering would be noninfringing unless they were substantially identical to the reverse engineered chip. See 130 CONG. REC. 28,960 (1984) (explanatory memorandum to the Mathias-Leahy Amendment to S. 1201). Section 906 differs from patent rules in two significant respects: first, it creates a right in unlicensed firms to engage in intermediate copying of the protected innovation, which patent law does not, and second, it allows the new product resulting from reverse engineering to be free from blocking intellectual property rights, as would generally be the case with patents as to subsequent inventions substantially incorporating the innovator’s invention.

⁹⁶ See, e.g., E.J. Chikofsky & J.H. Cross, *Reverse Engineering and Design Recovery: A Taxonomy*, 7 IEEE Software 13 (1990) (defining forward engineering); Kasch, supra note 84, at 85 (discussing forward engineering in respect of the SCPA).

⁹⁷ See supra Section II-A.

⁹⁸ Kasch, supra note 88, at 78.

⁹⁹ Id.

Semiconductor firms have historically relied on lead-time and secrecy far more than on patents to protect their intellectual assets.¹⁰⁰ An innovator could rely not only on being first to market to provide some lead-time, but also on being further along the yield curve than imitating second comers.¹⁰¹ Trade secrecy protection was especially important in the chip manufacturing process because considerable know-how was required to make commercially acceptable chips. However, trade secrecy law could obviously not protect the layout of chips sold in the marketplace because this information was readily ascertainable from examination of the marketed product (that is, it could be readily reverse engineered).¹⁰²

Several factors contributed to patents not playing a crucial role in the early and mid-phases of this industry.¹⁰³ For one thing, semiconductors are a cumulative system technology in which the interrelatedness of inventions required extensive cross-licensing of patents in order for industry participants to make advanced chips.¹⁰⁴ Second, some major customers of this industry, including notably the U.S. government, insisted on “second-sourcing,” that is, in having competitive suppliers of compatible chips to reduce the risk of unforeseen supply problems.¹⁰⁵ This too contributed to widespread cross-licensing. Third, the rapid pace of innovation and short life cycles of many chip products lessened the utility of patents in this industry.¹⁰⁶ Fourth, during the 1970’s, when the semiconductor industry was becoming a major American industry, there was a widespread perception that courts were hostile to patents, and patents had, as a consequence, less economic significance than at other times.¹⁰⁷ A fifth limitation of patents, much emphasized in the legislative history of SCPA, was that under then prevailing standards, the overall layout of chip circuits was rarely if ever patentable.¹⁰⁸

¹⁰⁰ See, e.g., Cohen et al., supra note 45, Tables A1 and A2 (showing that semiconductor firms regard trade secrecy and lead-time as far more effective patents in protecting firm intellectual assets from market-destructive appropriations). See also Levin et al., supra note 53.

¹⁰¹ Early stages of a chip production process generally results in a lower yield of salable chips than in later stages when fine-tuning of the production process yields a higher quantity of salable chips. See, e.g., Kastenmeier & Remington, supra note 73, at 452-53

¹⁰² See, e.g., Reichman, *Legal Hybrids*, supra note 14, at 2478-80.

¹⁰³ See, e.g., Bronwyn H. Hall and Rosemarie H. Ziedonis, *The Patent Paradox Revisited: An Empirical Study of Patenting in the US Semiconductor Industry, 1979-95*, 32 *Rand J. Econ.* 101 (May 2000), Table B.2 (showing the pattern of patenting in the semiconductor industry over this period).

¹⁰⁴ See, e.g., Suzanne Scotchmer, *Standing on the Shoulders of Giants: Cumulative Innovation and the Patent Law*, 5 *J. Econ. Persp.* 29 (1991) (discussing cumulative system technologies); Robert P. Merges & Richard Nelson, *On the Complex Economics of Patent Scope*, 90 *Colum. L. Rev.* 839 (1990) (same); Hall & Ziedonis, supra note 97, at 3 (characterizing semiconductor industry as cumulative system technology and emphasizing importance of cross-licensing in this industry); Deepak Somaya & David J. Teece, *Combining Patented Inventions in Multi-Invention Products: Transactional Challenges & Organizational Choices*, draft dated Aug. 24, 2001 (on file with the authors). See also Risberg, supra note 90, at 249 (noting widespread licensing in semiconductor industry).

¹⁰⁵ See, e.g., id, at 247, n. 29; Kasch, supra note 89, at 96-98 (discussing second sourcing).

¹⁰⁶ See, e.g., Hall & Ziedonis, supra note 104, at 2. See also Goldberg, supra note 87, at 330.

¹⁰⁷ See, e.g., Risberg, supra note 89, at 267. As patents grew progressively stronger in the 1980’s, chip firms increased the rate of their patenting. See id. at 267-79; Hall & Ziedonis, supra note 104, at 4.

¹⁰⁸ See, e.g., H. Rep. No. 98-781, 98th Cong., 2nd Sess. at 3-4 (May 15, 1984).

While the U.S. semiconductor industry thrived for years under these conditions, the life cycle pattern of chip products was so disrupted during the late 1970's and early 1980's that leading chip producers sought legislative help. Several factors contributed to this disturbance. One was a steep rise in the cost of developing and marketing new chips.¹⁰⁹ Second, advances in chip manufacturing technologies dramatically reduced the cost and time required to make exact or near-exact competing chips, thereby shortening considerably the lead-time innovators could expect and reducing the costs of copying.¹¹⁰ Third, American firms were losing out to foreign—and in particular, to Japanese—competitors, raising the specter of a diminished U.S. presence in this very significant sector of the national and global economy with potentially serious national security consequences.¹¹¹

B. Copyright or Sui Generis Protection for Chip Designs?

Intel Corp. initially sought to combat “chip piracy” with copyright law. It obtained copyright registration certificates for drawings of chip circuitry,¹¹² and then sought to register masks (that is, stencils used in manufacturing chips) and chips as derivative works of the drawings. This would provide a basis for claiming that manufacturers of identical or near-identical chips were infringing copyrights in protected drawings, masks and/or chips. Intel's strategy was derailed when the U.S. Copyright Office rejected Intel's application to register chips because of their utilitarian function.¹¹³ Although Intel sued the Register of Copyright to compel registration,¹¹⁴ it soon dropped the litigation and turned to Congress for legislative relief.¹¹⁵

Intel's second strategy was also based on copyright. It asked Congress to amend copyright law to add “mask works” to the subject matter of copyright.¹¹⁶ Intel argued that innovative chip designs, like literary works, were very expensive to develop and very cheap to copy, and unless the law intervened to stop rapid, cheap copying, innovators would be unable to recoup their R&D expenses and justify further investments in semiconductor innovation.¹¹⁷ A nearly identical argument was made by a Congressional

¹⁰⁹ See, e.g., Kasch, supra note 89, at 78-79 (estimating costs of new chip development at \$40-50 million by 1983).

¹¹⁰ See id. (estimating the costs of chip cloning at \$50,000-\$100,000 which could be done in three to six months).

¹¹¹ See, e.g., id. at 79. See also Raskind, supra note 94, at 385, 413-15.

¹¹² See Kasch, supra note 89, at 80.

¹¹³ Id. See also Statement of Dorothy Schrader, Associate Registrar of Copyrights (cited hereinafter as “Schrader Statement”), House Hearings, supra note 88, at 87-88 (questioning the registrability of masks because of their role in the process of manufacturing chips).

¹¹⁴ See, e.g., id. at 88, n. 10.

¹¹⁵ Stern gives a chronology of the legislative activity on the chip bills in Stern, supra note 92, in Appendix B. He reports that H.R. 14293, 95th Cong., 2d Sess. (1978) was the first bill introduced in Congress to protect chip designs through copyright law. The same bill was reintroduced the next session as H.R. 1007, 96th Cong., 1st Sess. (1979), and hearings were held on it, but no action was taken. Similar bills were introduced in the 97th Congress, but it was not until the 98th Congress that there was sufficient consensus on semiconductor chip protection for the legislation move forward and pass. See Stern, supra note 92, at 493-95.

¹¹⁶ See Kasch, supra note 89, at 80.

¹¹⁷ See, e.g., Dunlap Statement, supra note 88.

Commission on New Technological Uses of Copyrighted Works (CONTU) to support use of copyright law to protect computer programs.¹¹⁸ Because programs and chips are both utilitarian information technology products that are expensive to develop and cheap and easy to copy, one might have thought that copyright should be used for both or for neither. Yet, the copyright argument was successful as to programs,¹¹⁹ although not as to chips.

During the first set of legislative hearings on the chip protection bills, some industry witnesses expressed concern about the use of copyright for chips or mask works because copyright's fair use doctrine seemed too uncertain a basis for ensuring that the common and competitively healthy industry practice of reverse engineering could continue.¹²⁰ An explicit reverse engineering privilege was added to a later bill. However, it allowed reproducing a chip design for study and analysis without expressly allowing reverse engineers to use the results in designing a new chip.¹²¹ Industry representatives pointed out that in order to comply with second-source "form, fit, and function" compatibility requirements, the chips resulting from reverse engineering would necessarily be quite similar to the chips being reverse engineered, although not in a competitively harmful way.¹²²

Lack of industry consensus stalled movement on chip protection bills until 1983. By that time, a fairly large number of compromise provisions had been added to the bills to satisfy various semiconductor industry concerns.¹²³ Yet those compromises so deviated from traditional copyright rules that a new and different kind of opposition arose.¹²⁴ As a representative of the Association of American Publishers explained at a 1983 hearing:

[T]he AAP is not questioning the creativity, skill, labor, or investment of chip designers, or their need for and entitlement to appropriate protection...Our concern lies...with the fundamental departures from the copyright system that accompany the proposal, e.g., the extension of Copyright Act protection to utilitarian objects that, it is acknowledged, may not be 'writings' under the Constitution...; the limitations on remedies against infringement and the extension of compulsory licensing;

¹¹⁸ See National Commission on New Technological Uses of Copyrighted Works, Final Report 12-13 (1978) (cited hereinafter as "CONTU Report"). But see Pamela Samuelson, *Creating a New Kind of Intellectual Property Law: Applying the Lessons of the Chip Law to Computer Programs*, 70 Minn. L. Rev. 471, 504-06 (1985) (arguing that the Congressional rationale for protecting chip designs by means of a sui generis law suggested that computer programs should also have been protected through a sui generis law).

¹¹⁹ See Pub. L. No. 96-517, 94 Stat. 3015, 3028 (1980) (implementing CONTU's recommendations for amending copyright law to protect programs).

¹²⁰ See, e.g., Kasch, *supra* note 89, at 81 (reporting sharp industry divide at first hearing on chip legislation).

¹²¹ *Id.* at 82.

¹²² See, e.g., Brown, *supra* note 94, at 998-99.

¹²³ See, e.g., Kasch, *supra* note 89, at 81.

¹²⁴ See, e.g., S. 1201, 98th Cong., 1st Sess. (1983), discussed Schrader Statement, *supra* note 113, at 128-33 (comparing the main features of the sui generis and copyright bills).

and most notably, the limitation imposed on the duration of this particular class, and the distortion of the fair use doctrine to accommodate reverse engineering.¹²⁵

It would be better, he argued, to develop “sui generis” (of its own kind) legislation to protect semiconductor chip designs—which is what Congress ultimately did in 1984.¹²⁶

The SCPA regime resembles copyright in significant respects.¹²⁷ One conceptual holdover from Intel’s copyright strategy was the subject matter chosen for SCPA protection, namely, “mask works.”¹²⁸ As with copyright, mask works must be “original” to qualify for protection.¹²⁹ Rights attach automatically by operation of law, but registration with the Copyright Office brings benefits unavailable to non-registrants.¹³⁰ The legislative history demonstrates that copyright-like concepts of substantial similarity and substantial identity were to be used in judging infringement of SCPA rights.¹³¹ And SCPA relies, as copyright does, on a grant of exclusive rights to control reproductions and distributions of products embodying the protected work.¹³²

¹²⁵ See Testimony of Jon A. Baumgarten, Copyright Counsel, Association of American Publishers, House Hearings, *supra* note 88, at 11-12. See also *id.* at 12, n. 2 (expressing doubt that reverse engineering would be fair use under traditional principles of that law).

¹²⁶ *Id.* at 11.

¹²⁷ See, e.g., Reichman, *Legal Hybrids*, *supra* note 14, at 2478-79 (discussing similarities between SCPA and copyright).

¹²⁸ 17 U.S.C. sec. 902. In retrospect, it would have been preferable for the subject matter of SCPA protection to be the layout, design, or topography of integrated circuits. Subsequent legislation in other countries has chosen the topography of integrated circuits as its subject matter. See, e.g., Council Directive of 16 December 1986 on the Legal Protection of Topographies of Semiconductor Products, 87/54/EEC, 30 O.J. L24 at 36 (Jan. 27, 1987) (cited hereinafter as “Council Directive”). A serious disadvantage of “mask works” as the protected subject matter under SCPA is that its technology-specific nature meant that SCPA would become obsolete if chip production moved beyond use of masks in the manufacturing process—as indeed occurred. See Goldberg, *supra* note 87, at 333.

¹²⁹ See 17 U.S.C. sec. 902(b)(1). The SCPA denies protection to chip designs that are “staple, commonplace, or familiar in the semiconductor industry, or variations of such designs in such a way that, considered as a whole, is not original.” *Id.* at 902(b)(2). However, Congress offered very little guidance about the quantum of originality required for SCPA protection or how much difference must exist between the second comer’s and the innovator’s chips before subsequent chips will be deemed noninfringing. See, e.g., Brown, *supra* note 94, at 991-92; Risberg, *supra* note 90, at 262.

¹³⁰ 17 U.S.C. sec. 908 (rights under SCPA terminate unless the chip design is registered within two years). See also 17 U.S.C. sec. 412 (right to statutory damage awards and to recovery of attorney fees depends on prompt registration of copyright claims with the Copyright Office).

¹³¹ See, e.g., H. R. Rep. No. 781, 98th Cong., 2d Sess., at 18, 22 (anticipating use of copyright-like concepts of substantial similarity and substantial identity in infringement decisions). However, second comers cannot hope to make a workable compatible chip merely by making minor variations on an innovative chip design in order to avoid infringement. As one commentator has noted, “[v]ery subtle variations in logic flow, or in certain arrangement configurations, may make interchangeability impossible.” Brown, *supra* note 94, at 998.

¹³² Compare 17 U.S.C. sec. 905 (SCPA’s exclusive rights provision) and 17 U.S.C. sec. 106 (copyright’s exclusive rights provision). One very significant difference between the exclusive rights provision of the SCPA and that of copyright is that the former does not include a derivative work right.

A notably sui generis feature of the SCPA¹³³ is its reverse engineering provision:

[I]t is not an infringement of the exclusive rights of the owner of a mask work for (1) a person to reproduce the mask work solely for the purpose of teaching, analyzing, or evaluating the concepts or techniques embodied in the mask work or the circuitry, logic flow, or organization of components used in the mask work; or (2) a person who performs the analysis or evaluation described in paragraph (1) to incorporate the results of such conduct in an original mask work which is made to be distributed.¹³⁴

Industry witnesses distinguished between “legitimate” and “illegitimate” reverse engineering:

A reverse engineering firm should be allowed to analyze the chip, draw a circuit schematic of the chip, and then lay out a different pattern. This pattern could be used to fabricate a version of the semiconductor which is functionally equivalent to the original chip but has different visual patterns on it. The reverse engineering firm could then improve the performance of the chip, reduce the size of the chip, and reduce the overall manufacturing costs of the chip....¹³⁵

A “legitimate” reverse engineer would not, for example, reproduce inefficiencies or mistakes in the innovator’s layout of circuits because careful study and analysis of the chip would identify these problems.¹³⁶

The House Report on SCPA explained the impact of this and similar testimony:

Based on testimony of industry representatives that it is an established industry practice to...make photo-reproductions of the mask work in order to analyze the existing chip so as to design a second chip with the same electrical and physical performance characteristics as the existing chip (so-called ‘form, fit and function’ compatibility), and that this practice fosters fair competition and provides a frequently needed ‘second source’ for chip products, it is the intent of the Committee to permit such reproduction by competitors...[and to make illegal] mere wholesale appropriation of the work and investment in the first chip. It is the intent of the Committee to permit, under the reverse engineering limitation, the...creation of a second

¹³³ The SCPA contains a number of novel and specially tailored provisions apart from the reverse engineering privilege. See, e.g., Samuelson, supra note 118, at 492-501 (discussing other sui generis features of SCPA).

¹³⁴ 17 U.S.C. sec. 906(a). Similar provisions exist in other laws protecting chip designs. See, e.g., Council Directive, supra note 128, Art. 5(2)-(4).

¹³⁵ Dunlap Statement, supra note 88, at 27-28.

¹³⁶ Id. Some industry witnesses also sought to distinguish “legitimate” from “illegitimate” reverse engineering in terms of differences in comparative development costs and time to market, see, e.g., id., at 28, 32, or in terms of the “paper trail” that a legitimate reverse engineer would create, see id. at 36. The legislative history gives no weight to these factors. See, e.g., House Report, supra note 131.

mask work whose layout, in substantial part, is similar to the layout of the protected mask work—if the second mask work as the product of substantial study and analysis and not the mere result of plagiarism accomplished without such study or analysis.¹³⁷

One commentator characterized the SCPA as “accept[ing] copying as the industry norm of competition. The industry spokespersons, while seeking protection from piracy as they perceived it, were insistent on preserving and encouraging the industry practices of creative copying, a practice known to them as reverse engineering.”¹³⁸

C. An Economic Rationale for the SCPA Rules

Section II argued that reverse engineering does not unduly undermine incentives to invest in innovation as long as it is costly, time-consuming, or both. During the time that the SCPA and predecessor bills were pending in Congress, reverse engineering of chips could be done very cheaply and quickly by peeling away layers of a purchased chip, one at a time, photographing each layer, making a mask from these photographs, and then using these masks to manufacture identical chips.¹³⁹ The SCPA rules made this cheap and rapid route to competitive entry illegal and required reverse engineers to design original chips to avert infringement liability. The forward-engineering requirement lengthened second comers’ development time and increased their costs, thereby giving the innovator more lead-time within which to recoup its R&D expenses and more protection against clone-based pricing. The forward-engineering requirement also increased the likelihood that second comers would advance the state of the art in semiconductor design.¹⁴⁰ As long as second comers had to make their chips different, they might as well make them better as well.

Table 2 uses the same social welfare criteria as in Table 1 to illustrate our assessment of the economic effects of pre-SCPA rules as compared with post-SCPA rules.

Table 2
Social Calculus of Reverse Engineering in the Chip Industry Pre- and Post-SCPA

	Pre-SCPA	Post-SCPA
Incentives to innovate	lower (too low)	higher

¹³⁷ Id. at 22.

¹³⁸ Raskind, supra note 94, at 391. Shortly after the enactment of SCPA, Professor Raskind predicted that “[w]hen Congress introduced the concept of ‘reverse engineering’ as a limitation on the rights of an owner of protected industrial intellectual property in the Semiconductor Chip Protection Act of 1984 (‘the Chip Act’), it effected an innovation in the law of intellectual property that has ramifications wider and deeper than the Chip Act itself.” Id. at 385. As Section IV-A will show, this prediction has proved accurate.

¹³⁹ See H. Rep., supra note 131.

¹⁴⁰ See, e.g., Ted O’Donoghue & Suzanne Scotchmer, & Jacques-Francois Thisse, *Patent Breadth, Patent Life and the Pace of Technological Progress*, 7 J. Econ. & Mgmt. Strategy 1 (Spring 1998)(discussing the effective life of intellectual property protection in rapidly evolving sequential technologies and how the breadth of protection interacts with this).

Price	lower	higher
Follow-on innovation	lower (too low)	higher
Socially wasteful costs	lower	higher (but avoidable by licensing)

Incentives to invest in innovative chip designs were too low before enactment of SCPA because cloners rapidly eroded lead-time advantages for innovators. Incentives to innovate were restored once cloning was no longer an option. Incentives to invest in follow-on innovation were also very low in the pre-SCPA era because firms capable of investing in improved chips chose instead to clone while it was still legal. When chip cloning became illegal, firms had strong incentives to invest in improvements. Although consumers may have initially benefited from lower prices in the pre-SCPA era, prices were so low that innovators couldn't recoup their costs. SCPA may result in more socially wasteful costs because some second comers may spend resources making chip circuitry different to satisfy the originality requirements. However, some of these wasted costs are avoidable by licensing.

From an economic standpoint, the anti-cloning rules of the SCPA are designed to achieve much the same result as the anti-plug mold rules discussed in Section II, although they do so by a different technique. Chip cloners were no more engaged in innovation-enhancing discovery of applied industrial know-how than were plug-molders. The SCPA rule inducing second comers to join the ranks of innovation-enhancing firms is similar to the anti-plug mold rule that induced second comers to engage in more conventional forms of reverse engineering likely to advance the state of the art of boat hull design. SCPA achieves this result by establishing a kind of "breadth" requirement for subsequent products in contrast to the anti-plug mold laws that instead outlawed a particular means for making a competing product.¹⁴¹

D. Post-SCPA Developments

¹⁴¹ The notion of "breadth" has no formal meaning in law. However, the economics literature has interpreted breadth as measuring how much a product must be improved to avoid infringing a prior patent. See, e.g., *id.* See also Jerry Green & Suzanne Scotchmer, *On the Division of Profit between Sequential Innovators*, 26 RAND J. Econ. 20 (Spring 1995). In patent law, the requirements of nonobviousness and novelty jointly govern both the "breadth" of a patent, and the advance over prior art required for "patentability." These requirements are joined in SCPA, so that any improvement either escapes infringement and receives protection as a joint package, or does not. In general the requirements to escape infringement and receive protection are different. See Mark A. Lemley, *The Economics of Improvement in Intellectual Property*, 75 Tex. L. Rev. 989 (1997) for a discussion of copyright law and patent law in this regard. We emphasize breadth in our discussion of SCPA because SCPA solved the problem of cloning by making sure that clones infringe, and solved the problem of encouraging improvement by allowing the reverse engineer to escape infringement by improving the chip. See *infra* Sec. VI-A(2) (discussing breadth).

There has been very little litigation under the SCPA rules. Yet, the one reported judicial decision under SCPA is instructive because it involved a failed reverse engineering defense. In *Brooktree Corp. v. Advanced Micro Devices, Inc.*,¹⁴² AMD produced a prodigious paper trail in support of its reverse engineering defense and pointed to the considerable time and expense it had spent on developing a chip compatible with the Brooktree chip.¹⁴³ It also emphasized many differences in the layout of its chip circuitry and Brooktree's.¹⁴⁴ However, under pressure of an impending deadline, AMD's principal designer revisited the Brooktree chip layout and thereafter abandoned his plans for a six or eight transistor core cell design in favor of the same ten transistor design arrangement in Brooktree's chip.¹⁴⁵ The Court of Appeals concluded that a reasonable jury "could have decided that AMD's paper trail, insofar as it related to the SRAM cell, related entirely to AMD's failures, and that as soon as the Brooktree chip was correctly deciphered by reverse engineering, AMD did not create its own design but copied the Brooktree design."¹⁴⁶ While AMD surely made a far greater investment in engineering than the cloning firms that the SCPA was principally aimed at, AMD did not, as the SCPA required, develop its own original design of a key portion of the Brooktree chip, and hence, it was held liable for infringement of the SCPA right.

One way to interpret the scarcity of litigation under SCPA is as a sign that the law successfully deterred chip piracy. However, most legal commentators have inferred from this that the SCPA is unimportant.¹⁴⁷ Some put the blame on bad drafting, claiming that SCPA is technologically obsolete or provides too thin a scope of legal protection.¹⁴⁸ Others assert that SCPA became unimportant because of subsequent legal developments, such as the renewed importance of patents in the aftermath of creation of the Federal Circuit or the rise of second-source licensing agreements between pioneers and follow-on innovators.¹⁴⁹ Still others assert that technological changes, such as further miniaturization of chip circuitry, advances in process technology, mass customization of chip designs, and the increasing sophistication of CAD/CAM programs for generating that alternative layouts, rendered infeasible the kind of copying that gave rise to the SCPA.¹⁵⁰

¹⁴² 977 F.2d 1555 (Fed. Cir. 1992).

¹⁴³ *Id.* at 1566-67.

¹⁴⁴ *Id.* at 1566-67.

¹⁴⁵ *Id.* at 1567.

¹⁴⁶ *Id.* at 1568.

¹⁴⁷ See, e.g., Risberg, *supra* note 90, at 245 (describing SCPA as "a largely untested, if not impotent, piece of legislation"); Kasch, *supra* note 89, at 72 (SCPA of "largely academic interest").

¹⁴⁸ See, e.g., Goldberg, *supra* note 87, at 333-35 (making both complaints).

¹⁴⁹ See, e.g., See Hall & Ziedonis, *supra* note 103, at 4 (attributing a substantial increase in patenting in the semiconductor industry to a strong "pro-patent" shift in U.S. legal environment).

¹⁵⁰ See, e.g., Risberg, *supra* note 90, at 263-72; Kasch, *supra* note 89, at 73, 103. Kasch predicted that further changes in technology might cause the SCPA's anti-cloning protection to have renewed importance in the future. *Id.* at 103-04.

One indication of a continuing interest in SCPA among chip designers have can be found in the number of chip designs registered with the U.S. Copyright Office and counterpart agencies elsewhere.¹⁵¹ Legal protection for the layout of integrated circuits was also deemed important enough to warrant its inclusion in the TRIPS Agreement.¹⁵² TRIPS incorporates by reference a number of provisions of an earlier treaty on the legal protection for the layout of integrated circuits, including a reverse engineering privilege closely modeled on the SCPA rule.¹⁵³ The semiconductor chip industry, as a consequence, is the only industry whose reverse engineering activities are expressly protected in an international intellectual property treaty.

In the years since the SCPA enactment, the semiconductor industry has enjoyed very considerable growth and U.S. firms have dominated a much-enlarged global chip market.¹⁵⁴ Interestingly, in the post-SCPA era, there has been a partial bifurcation of design and fabrication components of the chip industry.¹⁵⁵ That is, some firms now design chip layouts and other firms fabricate chips of that design. This has been accompanied by a rise in the rate of patenting

¹⁵¹ See, e.g., Risberg, *supra* note 89, at 243, n. 16. See also Andy Y. Sun, *From Pirate King to Jungle King: Transformation of Taiwan's Intellectual Property Protection*, 9 *Fordham Intell. Prop., Media & Ent. L.J.* 67, 138-39 (1998) (reporting a substantial number of chip protection registrations in Taiwan). Professor Rosemarie Ziedonis has collected data about chip registrations in the U.S. She reports that between 1985 and 1997, there were 6834 chip registrations with the U.S. Copyright Office, including 637 in 1996 and 471 in 1997. Ironically, Intel is noticeably absent from the list of U.S. registrants. Email communication from Rosemarie Ziedonis, May 18, 2001 (on file with author).

¹⁵² See TRIPS, *supra* note 3, arts. 35-38. On the subject of international protection for chip designs, it is worth noting that the United States made what in retrospect can be seen as a tactical mistake in its approach to gaining international acceptance of SCPA-like protection. Rather than adopt a national treatment-based approach, as most international treaties do, under which chip designs of foreign producers would be protected under U.S. law regardless of whether their nations protected chip designs, SCPA adopted a material reciprocity approach under which the chips of foreign nationals would not be protected under U.S. law unless their nations had adopted "equivalent" laws. SCPA established a process under which U.S. officials could judge whether other nations had adopted sufficient laws. See 17 U.S.C. sec. 914. Although the U.S. was able to persuade many other nations to adopt chip protection laws, see, e.g., Council Directive, *supra* note 128, and Stern, *supra* note 92, chap. 10, there has been some resentment among intellectual property professionals in other countries about the U.S. reciprocity approach. This also came back to haunt the U.S. when the European Commission decided to adopt a new form of legal protection to the contents of databases on a material reciprocity basis. This was of concern to U.S. database developers because of their substantial market share in the European market. See, e.g., J.H. Reichman & Pamela Samuelson, *Intellectual Property Rights in Data?*, 50 *Vand. L. Rev.* 51, 96-97 (1997)(discussing the European database directive and the initial U.S. objections to its reciprocity provision).

¹⁵³ See TRIPS, *supra* note 3, art. 35 (incorporating various provisions of the Treaty on Intellectual Property in Respect of Integrated Circuits, commonly known as the "Washington Treaty," including Art. 6(2)'s reverse engineering privilege). The U.S. and other developed chip-producing countries objected to some provisions of the Washington treaty and refused to sign it. See, e.g., Goldberg, *supra* note 87, at 335-36 (discussing various dissatisfactions with the Washington Treaty). To "fix" the perceived weaknesses in the Washington Treaty, the TRIPS Agreement added some new substantive requirements as minimum standards for protecting the layouts of integrated circuits. See TRIPS, *supra* note 3, arts. 36-37.

¹⁵⁴ See, e.g., Risberg, *supra* note 90, at 273-76; Hall & Ziedonis, *supra* note 103. See generally U.S. DEPT. OF COMMERCE, *THE EMERGING DIGITAL ECONOMY*, Appendix 1 (1998) (reporting on high growth of information technology industries, including the semiconductor industry, in the U.S.).

¹⁵⁵ See, e.g., Hall & Ziedonis, *supra* note 103, at 4-5.

in this industry and a more aggressive enforcement of patent rights, especially by the design firms.¹⁵⁶ From an economic perspective, if SCPA contributed to the rise in second-source licensing agreements (and it probably did) and if it contributed to the cessation of cloning of innovative chip designs, it had a beneficial effect on this market.

IV. Reverse Engineering in the Computer Software Industry

Reverse engineering is as standard an industry practice in the computer software industry as in the traditional manufacturing and semiconductor industries.¹⁵⁷ However, for much of the past two decades, the legality of two common forms of software reverse engineering, namely, decompilation and disassembly of object code,¹⁵⁸ has been challenged on trade secret, copyright, patent, and contract law theories. This section will first review the legal debate about reverse engineering of computer software as a matter of intellectual property law and explain why courts and legal commentators have overwhelmingly supported the legality of such reverse engineering. It will then go on to assess the economic effects of decompilation and disassembly of program code, particularly when done for purposes of developing a program capable of interoperating with another program. The economic case for allowing reverse engineering to achieve interoperability is not as open and shut as some legal commentators have suggested.¹⁵⁹ However, we believe that interoperability has, on balance, more beneficial than harmful economic consequences. Hence, a legal rule permitting reverse engineering of programs to achieve interoperability is economically sound. This section concludes with a discussion of the legal debate over enforceability of contractual restrictions on reverse engineering of computer software and economic reasons for not enforcing them.

A. Reverse Engineering of Software And Copyright Law

Commercial developers of computer programs generally distribute software in object code form. They do so for two principal reasons: first, because users mainly want the functionality that object code forms of programs provide, not to read the program's text, and second, because the developers want to maintain source code forms of their products and other human-readable documentation as trade secrets.¹⁶⁰ Decompilation or

¹⁵⁶ See, e.g., *id.* One would expect design firms to rely not only on patents (as they apparently do, see *id.* at 5), but also on the legal protection SCPA provides against copying of chip layouts, although the latter has not been documented.

¹⁵⁷ See, e.g., Andrew Johnson-Laird, *Reverse Engineering of Software: Separating Legal Mythology From Actual Technology*, 5 *Software L.J.* 331, 354 (1992) (“Reverse engineering is practiced by all programmers....”).

¹⁵⁸ See OFFICE OF TECHNOLOGY ASSESSMENT, U.S. CONGRESS, FINDING A BALANCE: COMPUTER SOFTWARE, INTELLECTUAL PROPERTY, AND THE CHALLENGE OF TECHNOLOGICAL CHANGE 7 (1992)(explaining disassembly and decompilation).

¹⁵⁹ See *infra* note 187.

¹⁶⁰ See, e.g., Jessica Litman, *Copyright and Information Policy*, 55 *Law & Contemp. Probs.* 185, 196-201 (1992) (discussing strategies of software industry lawyers for maintaining program internals as trade secrets). See also Reichman, *Programs as Know-How*, *supra* note 1, at 701 (describing nondisclosure of program internals as a business imperative, although concluding that second comers ought to be able to reverse engineer object code).

disassembly of object code provides a way for reverse engineers to “work[] backwards from object code to produce a simulcrum of the original source code.”¹⁶¹ From this approximation of source code, reverse engineers can discern or deduce internal design details of the program, such as information necessary to develop a program that will interoperate with the decompiled or disassembled program. Lawyers for some major software producers argued that decompilation and disassembly should be illegal as a matter of copyright and trade secrecy law. They argue that the unauthorized copies of programs made in the process of decompiling or disassembling them infringe the program copyright, and this infringement makes the decompilation or disassembly an improper means of obtaining program trade secrets.¹⁶²

¹⁶¹ See Cohen & Lemley, *supra* note 12, at 16, n. 52 (2001). See also Litman, *supra* note 160, at 197-98: Decompilation is a species of reverse engineering that involves translating the object code into a human-readable form, or ‘pseudo source code,’ largely through trial and error. Part of the decompilation process can be computer-assisted: there are, for example, disassembly programs that will translate the object code into an intermediate assembly language form that is more decipherable to skilled readers. Other computer software can assist the developer in the laborious process of translating the assembly language into pseudo source code form. The decompilation process does not generate source code as originally written but rather a plausible reconstruction of what portions of the original source code could have been. Of course, the produce of such reverse engineering will include not only the parts of the program that were compiled into object code in the first instance; the English comments and descriptions were never compiled and cannot be retrieved or recreated. Pseudo source code is nonetheless a useful tool that can assist a software developer in analyzing how a computer program works.

¹⁶² See, e.g., Allen R. Grogan, *Decompilation and Disassembly: Undoing Software Protection*, Computer Law., Jan. 1984, at 1. Grogan’s argument wove trade secret, copyright, and contract together in a tight mesh. He asserted that reverse engineering of object code by decompilation or disassembly was trade secret misappropriation because the reverse engineer used improper means to obtain the trade secret information embedded in the program by making unauthorized copies of the program in the course of the reverse engineering process (thereby infringing copyright) and/or by violating anti-reverse engineering clauses of shrinkwrap license contracts under which they were distributed. At that time, there was much uncertainty about the enforceability of shrinkwrap licenses as a matter of contract law and about the enforceability of anti-reverse engineering clauses in particular. See *infra* Section IV-D for further discussion of the shrinkwrap license issues pertaining to reverse engineering of software. For similar arguments, see also Anthony L. Clapes, *Confessions of an Amicus Curiae: Technophobia, Law and Creativity in the Digital Arts*, 19 U. Dayton L. Rev. 903 (1994); Duncan Davidson, *Common Law, Uncommon Software*, 47 U. Pitt. L. Rev. 1037 (1985); Arthur R. Miller, *Copyright Protection for Computer Programs, Databases and Computer-Generated Works: Is Anything New Since CONTU?*, 106 Harv. L. Rev. 977 (1993).

However, the predominant view among legal commentators supports a right to reverse engineer software under copyright law. See, e.g., JONATHAN BAND & MASANOBU KATOH, INTERFACES ON TRIAL: INTELLECTUAL PROPERTY AND INTEROPERABILITY IN THE GLOBAL SOFTWARE INDUSTRY 167-225 (1995); *Brief Amicus Curiae of Eleven Copyright Professors*, Sega Enter. Ltd. v. Accolade, Inc., 977 F.2d 1510 (9th Cir. 1992), published in 33 Jurimetrics J. 147 (1992); Julie E. Cohen, *Reverse Engineering and the Rise of Electronic Vigilantism: Intellectual Property Implications of ‘Lock-out’ Programs*, 68 S. Cal. L. Rev. 1091 (1995); Lawrence Graham & Richard O. Zerbe, Jr., *Economically Efficient Treatment of Computer Software: Reverse Engineering, Protection and Disclosure*, 22 Rutg. Comp. & Tech. L.J. 61 (1996); Dennis S. Karjala, *Copyright Protection of Computer Software, Reverse Engineering and Professor Miller*, 19 U. Dayton L. Rev. 975 (1994); Robert A. Kreiss, *Accessibility and Commercialization in Copyright Theory*, 43 UCLA L. Rev. 1 (1995); *LaST Frontier Conference Report on Copyright Protection for Computer Software*, 30 Jurim. J. 15 (1989); Ronald S. Laurie & Stephen M. Everett, *Protection of Trade Secrets in Object Form Software: The Case for Reverse Engineering*, Computer Law., July 1984, at 1; Lemley & McGowan, *supra* note 47; Litman, *supra* note 149;

The principal decision testing this legal theory was *Sega Enterprises Ltd. v. Accolade, Inc.*¹⁶³ Accolade, a small U.S. computer game company, disassembled Sega game programs in order to get information necessary to make its games compatible with the Sega Genesis console. Accolade then sold its independently developed games in competition with those made by Sega and third-party developers licensed by Sega. Accolade raised a fair use defense to Sega's claims that the disassembly copies were infringing.¹⁶⁴ The Ninth Circuit gave little weight to the commercial purpose of Accolade's copying because it regarded the copying as having been done "solely in order to discover the functional requirements for compatibility with the Genesis console— aspects of Sega's programs that are not protected by copyright."¹⁶⁵ Reverse engineering was, moreover, the only way that Accolade could gain access to this information.¹⁶⁶ Although Accolade had copied the whole of Sega's programs in the course of its reverse analysis, the court discounted this because it occurred in an intermediate stage of Accolade's software development process. Although the court recognized that Accolade's games affected the market for Sega games, it did not do so in a way in which copyright law is concerned.¹⁶⁷ Accolade's decompilation "led to an increase in the number of independently designed video game programs offered for use with the Genesis console. It is precisely this growth in creative expression...that the Copyright Act was intended to promote."¹⁶⁸ An important policy consideration was the court's recognition that if it ruled that disassembling computer programs was unlawful, this would confer on

Charles R. McManis, *Intellectual Property Protection and Reverse Engineering of Computer Programs in the United States and the European Community*, 8 High Tech. L. J. 25 (1993); Reichman, *Programs As Know-How*, supra note 1; David A. Rice, *Sega and Beyond: A Beacon for Fair Use Analysis...At Least As Far As It Goes*, 19 U. Dayton L. Rev. 1131 (1994); Pamela Samuelson, *Fair Use for Computer Programs and Other Copyrightable Works in Digital Form: The Implications of Sony, Galoob, and Sega*, 1 J. Intell. Prop. L. 49 (1993); Manifesto, supra note 14; Timothy Teter, Note, *Merger and the Machines: An Analysis of the Pro-Compatibility Trend in Computer Software Copyright Cases*, 45 Stan. L. Rev. 1061 (1993).

¹⁶³ 977 F.2d 1510 (9th Cir. 1992). *Sega v. Accolade* was not the first appellate court decision on whether decompilation or disassembly of a program could in appropriate circumstances be fair use. *Atari Games Corp. v. Nintendo of Am., Inc.*, 975 F.2d 832 (Fed. Cir. 1992) was decided shortly before the Ninth Circuit decision. The *Atari v. Nintendo* analysis of fair use is similar to the Ninth Circuit's analysis, although somewhat less extensive. In *Atari Games*, the fair use issue was complicated by the fact that Atari Games' lawyers lied to the U.S. Copyright Office to get the registration copy of Nintendo source code so that the firm's engineers could use it to finalize the development of compatible games. *Id.* at 837. The Federal Circuit ruled that the initial decompilation copying was fair use. *Id.* at 843.

¹⁶⁴ Courts generally consider four factors in considering whether a use is fair: the purpose of the defendant's use of the work, the nature of the copyrighted work, the amount and substantiality of the defendant's appropriation, and the harm or potential harm to the market if the defendant's use is permitted. See 17 U.S.C. sec. 107. It is interesting to note that Sega relied in part on the legislative history of SCPA in which some witnesses had expressed doubt that reverse engineering could be fair use as a matter of copyright law. See, e.g., Baumgarten Testimony, supra note 125 and accompanying text; *Sega*, 977 F.2d at 1517.

¹⁶⁵ 977 F.2d at 1522.

¹⁶⁶ *Id.* The unprotected aspects of most copyrighted works, the court pointed out, "are readily accessible to the human eye....Computer programs, however, are typically distributed for public use in object code form, embedded in a silicon chip, or on a floppy disk. For that reason, humans often cannot gain access to the unprotected ideas and functional concepts without disassembling that code." *Id.* at 1525.

¹⁶⁷ *Id.* Copyright law is concerned with infringing copies that compete with the author's works, not with competition on the merits among noninfringing works.

¹⁶⁸ *Id.* at 1523-24.

Sega “a de facto monopoly over [the unprotected] ideas and functional concepts [in the program].”¹⁶⁹ To get a monopoly on such ideas and functional concepts, a creator needs to seek patent protection.¹⁷⁰

Still, the court did not give a green light to all reverse engineering of program code, but only to that undertaken for a “legitimate purpose,” such as to gain access to the functional specifications necessary to make a compatible program, and then only if it “provides the only means of access to those elements of the code that are not protected by copyright.”¹⁷¹

The Ninth Circuit Court of Appeals recently reaffirmed *Sega v. Accolade* ruling in *Sony Computer Entertainment, Inc. v. Connectix Corp.*¹⁷² The main difference between it and *Sega v. Accolade* was that Connectix disassembled Sony programs in order to develop emulation software to allow owners of Apple computers to play Sony Playstation games on their iMacs. That is, Connectix reverse engineered in order to make a competing platform, not to make compatible games. The appellate court perceived no legal difference in the decompilation-for-interoperability considerations pertinent to development of competing platforms than as to games. In the wake of this loss, Sony has charged makers of emulation programs with patent infringement based on decompilation of its programs.¹⁷³ It will be interesting to see if the courts will be equally receptive to a decompilation-for-interoperability defense as a matter of patent law.¹⁷⁴

¹⁶⁹ Id. at 1527.

¹⁷⁰ Id. at 1526.

¹⁷¹ Id. at 1518.

¹⁷² 203 F.3d 596 (9th Cir. 2000)(reaffirming and extending *Sega v. Accolade* as to defendant who reverse engineered Sony games in order to develop software to enable users to play Sony games on Apple computers);

¹⁷³ See Cohen & Lemley, *supra* note 12, at 21. To illustrate how decompilation might run afoul of patent law, consider this variant on the *Sega v. Accolade* dispute: Assume that Sega had a patent on an algorithm used in all of its game programs. By disassembling Sega programs, Accolade would arguably “make” or “use” this patented aspect of Sega’s programs, even if it did so unconsciously and inadvertently.

¹⁷⁴ Cohen and Lemley have cogently argued for a limited reverse engineering privilege in patent law to allow decompilation of computer programs. Id. at 18-37. They point out that “because patent law contains no fair use or reverse engineering exemption, patentees could use the grant of rights covering a single component of a complex program to prevent any ‘making’ or ‘using’ of the program as a whole, including those temporary uses required for reverse engineering.” Id. at 6. They argue that “reverse engineering is an important means of preserving competition between products and of preserving compatibility between products. In markets characterized by network effects, such as software, this latter objective is particularly important.” Id. at 21. They also point out that “[r]everse engineering promotes the fundamental [patent] policies of disclosure and enablement, ensures that patents will not be leveraged to protect unprotectable components of software, preserves the balance sought by the intellectual property system as a whole, and also helps patentees to enforce their rights.” Id. at 22-23.

Cohen and Lemley consider various doctrines under which such a reverse engineering privilege might be established, including patent law’s experimental use defense, exhaustion of rights defense, implied license, and misuse. Id. at 29-36. They conclude that the policies underlying the exhaustion of rights and implied license doctrines of patent law should suffice to permit reverse engineering of programs. Id. at 32. If courts decide otherwise, Cohen and Lemley argue for legislation to permit it. Id. at 36-37. We agree that the limited reverse engineering rule they propose is legally and economically sound. See Maureen O’Rourke, *Towards a Fair Use Defense in Patent Law*, 100 Colum. L. Rev. 1177 (2000) (arguing for a fair use defense in patent law in part to enable decompilation for interoperability). It is worth pointing out that even before the Ninth Circuit *Sega v. Accolade* decision, the Court of Appeals for the Federal

Sega v. Accolade has been followed in virtually all subsequent cases.¹⁷⁵ It has been widely praised by legal commentators.¹⁷⁶ It is also consistent with the rules of other nations.¹⁷⁷ Those who argued that decompilation was and should be illegal predicted

Circuit had ruled that decompilation for purposes of interoperability could be a fair and noninfringing use of copyrighted programs. See *Atari Games*, 975 F.2d 832. Perhaps this augurs well for their recognition of a similar limited privilege as a matter of patent law, albeit on other than fair use grounds.

¹⁷⁵ See, e.g., *DSC Communications Corp. v. DGI Techs., Inc.*, 81 F.3d 597, 601 (5th Cir. 1996); *Bateman v. Mnemonics, Inc.*, 79 F.3d 1532, 1539 n.18 (11th Cir. 1996); *Mitel, Inc. v. Iqtel, Inc.*, 896 F.Supp. 1050, 1056-57 (D. Colo. 1995), *aff'd* on other grounds, 124 F.3d 1366 (10th Cir. 1997).

¹⁷⁶ See, e.g., sources cited *supra* note 162.

¹⁷⁷ The decompilation for interoperability issue was addressed legislatively in the European Union. In 1989, the European Commission published a proposed directive on the legal protection of computer programs to harmonize the laws of member states of the EU; it did not contain a decompilation or interoperability exception. See Proposal for a Council Directive on the Legal Protection of Computer Programs, 1989 O.J. (C91) 4. U.S. trade negotiators and representatives of some U.S. computer companies argued that this was as it should be. See, e.g., Victor Siber, *Interpreting Reverse Engineering Law*, IEEE Software 4 (July 1990) (explaining IBM's position against reverse engineering of software). See also Band & Katoh, *supra* note 162, at 229-241 (discussing U.S. industry lobbying and government officials' positions about the software directive); Thomas C. Vinje, *The Legislative History of the EC Software Directive*, in A HANDBOOK OF EUROPEAN SOFTWARE LAW (Michael Lehmann & Colin Tapper, eds. 1993). However, the Commission's competition directorate worried that unless the directive allowed decompilation for purposes of developing interoperable programs, European software developers would be at a serious disadvantage in the global software market. See, e.g., Pamela Samuelson, *Comparing U.S. and E.C. Copyright Protection For Computer Programs: Are They More Different Than They Seem?*, 13 J. Law & Comm. 279, 287-88 (1994) (discussing the concerns of the European Commission's competition directorate about the software directive).

In a response to these concerns, the final Directive contained a decompilation-for-interoperability privilege akin to that in *Sega v. Accolade*. See Council Directive 91/250 on the Legal Protection of Computer Programs, 1991 O.J. (L122) 42 (cited hereinafter as "Software Directive"). See, e.g., BRIDGET CZARNOTA & ROBERT J. HART, LEGAL PROTECTION OF COMPUTER PROGRAMS IN EUROPE: A GUIDE TO THE EC DIRECTIVE (1991); HANDBOOK OF EUROPEAN SOFTWARE LAW, *supra*. Achieving interoperability would seem to be the only legitimate purpose for decompilation under the European Software Directive. *Sega v. Accolade*, by contrast, contemplates that there may be other legitimate purposes for decompilation, although not saying what they might be. Error correction and detecting infringement are two other legitimate reasons to decompile programs. See, e.g., *E.F. Johnson Co. v. Uniden Corp. of Am.*, 623 F. Supp. 1485 (D. Minn. 1985) (decompilation to detect infringement); Samuelson, *supra* note 162, at 289 n. 59.

The European Software Directive also limits follow-on uses that can be made of information obtained in the course of decompilation. See European Software Directive, *supra*, Art. 6(2). One cannot, for example, publish information learned during reverse engineering. This puts at risk authors of books such as ANDREW SCHULMAN, DAVID MAXEY, & MATT PIETREK, UNDOCUMENTED WINDOWS: A PROGRAMMERS' GUIDE TO RESERVED MICROSOFT WINDOWS API FUNCTIONS (1992). Under Article 6(2), European decompilers are at risk if they try to recoup their reverse engineering expenses by licensing the information it learned in the course of its reverse engineering efforts. The Official Commentary to the Software Directive asserts that Article 6(2)(b) "prevents the publication or trafficking in information by those who have decompiled existing programs, since it would be inequitable to impose conditions on the decompiler but allows others access to the information which he had then made public." Czarnota & Hart, *supra*, at 81. The European software directive, in essence, converts copyright into a trade secrecy law as to internal elements of programs.

Europe's adoption of a decompilation-for-interoperability privilege and the *Sega v. Accolade* decision in the U.S. did not end the international debate about decompilation. U.S. officials continued to insist that decompilation should be unlawful. In the mid-1990's, for example, Japan was considering a proposal to amend its copyright law to allow reverse engineering of software, but dropped the proposal

grievous harm to the software industry if this form of reverse engineering was deemed lawful. These predictions have not been borne out. The American software industry has done well since 1992 when the *Sega v. Accolade* decision came down.¹⁷⁸

B. The Economics of Interoperability and Software Reverse Engineering

Sega v. Accolade and its progeny show that reverse engineering is undertaken for different reasons in the software industry than in other industrial contexts studied thus far. In manufacturing industries, reverse engineering is mainly undertaken in order to make directly competing stand-alone products.¹⁷⁹ Copyright law protects programs from the cheapest and most rapid way to make a directly competing identical product, namely, copying program code exactly.¹⁸⁰ However, reverse engineering of object code is generally so difficult, time-consuming, and resource-intensive that it is not an efficient way to develop competing but non-identical programs.¹⁸¹ As one technologist has explained, software reverse engineering does not “lay bare a program’s inner secrets. Indeed, it cannot. The inner secrets of a program, the real crown jewels, are embodied in the higher levels of abstraction material such as the source code commentary and the specification. This material never survives the process of being converted to object code.”¹⁸² A software reverse engineer must do considerable intellectual work to extract higher level abstractions and information from the text of the decompiled program, and still more work to incorporate what he or she has learned from this analysis in a new program.¹⁸³ In this respect, software resembles traditional manufacturing products because firms are generally insulated from market-destructive reverse engineering and

after intense pressure from U.S. officials. See, e.g., T.R. Reid, A Software Fight’s Blurred Battle Lines: U.S. Companies Are On Both Sides as Japan Considers Copyright Law Changes, Wash. Post, Jan. 11, 1994, at D1. However, some Japanese commentators believe that Japanese copyright law would permit decompilation for interoperability purposes. See, e.g., Band & Katoh, supra note 162, at 294-97; Keiji Sugiyama, *Reverse Engineering and Other Issues of Software Protection in Japan*, 11 Eur. Intell. Prop. Rev. 395 (1991). A number of jurisdictions have, however, adopted similar decompilation-for-interoperability exceptions to the European software directive. See, e.g., Band & Katoh, supra note 162, at 271-82.

¹⁷⁸ See, e.g., Contributions of the Packaged Software Industry to the Global Economy (April 1999) (study conducted by Pricewaterhouse Coopers, commissioned by the Business Software Alliance).

¹⁷⁹ See supra Section II-B.

¹⁸⁰ See, e.g., *Apple Computer v. Franklin Computer*, 714 F.2d 1240 (3d Cir. 1983)(exacting copying of Apple operating system programs infringed copyright).

¹⁸¹ See, e.g., Andrew Johnson-Laird, *Software Reverse-Engineering in the Real World*, 19 U. Dayton L. Rev. 843, 843 (1994). That is not to say that reverse engineering to make a directly competing product is unknown in the software industry, but it is uncommon. See, e.g., *Secure Services Techn., Inc. v. Time & Space Processing, Inc.*, 722 F.Supp. 1354 (E.D. Va. 1989)(reverse engineering of embedded software in secure facsimile machines for purposes of making competing, compatible facsimile machine); *Alcatel USA, Inc. v. DGI Techns., Inc.*, 166 F.3d 772 (5th Cir. 1999)(reverse engineering of telecommunications switching software to make competing product). Notice that both of these examples involve embedded software in a traditional manufactured product.

¹⁸² Johnson-Laird, supra note 181, at 896.

¹⁸³ It is worth noting that the nature of reverse engineering activities in the software industry is different than in manufacturing industries. Reverse engineering of manufactured products involves manipulation of physical objects. Reverse engineering of computer software involves analysis of program texts. See, e.g., Manifesto, supra note 15, at 2320: “Programs are machines whose medium of construction happens to be text.”

reimplementation because of high costs and difficulties of these activities for the purpose of making directly competing products.¹⁸⁴

Given the high costs and difficulties of software reverse engineering, it may seem surprising that it is such a standard industry practice. Software engineers reverse-analyze programs for a variety of reasons, including to fix “bugs” (i.e., logical errors in program instructions), to customize the program for the user’s needs (e.g., add some firm-specific features), to detect infringement, and to learn what others have done.¹⁸⁵ We focus our economic assessment of reverse engineering in the software industry on interoperability for two reasons: first, because this has been the most economically significant reason for software reverse engineering and second, because most of the litigation about software reverse engineering has involved interoperability issues.¹⁸⁶ As will become apparent, the economics of interoperability are more complicated than some previous commentators have suggested.¹⁸⁷

¹⁸⁴ This has caused some commentators to conclude that “decompilation should be regulated by the law—although not necessarily by copyright law—only if and to the extent that it permits competitors to acquire behavioral equivalence [with the target program] with only trivial effort, and therefore induces market failure.” See Manifesto, *supra* note 15, at 2392. Because the present state of decompilation technology does not permit trivial acquisition of equivalence, the Manifesto authors concluded that there is no economically sound reason to regulate decompilation. *Id.* If technological change shifted the balance and enabled rapid inexpensive copying that would be market-destructive, decompilation might need to be regulated to some degree. *Id.* at 2392-93. But see COMPUTER SCIENCE & TELECOMMUNICATIONS BOARD, NATIONAL RESEARCH COUNCIL, INTELLECTUAL PROPERTY ISSUES IN SOFTWARE 78 (1991) (quoting IBM executive as expressing concern that reverse analysis of programs could allow illegal copying of program internals that would escape easy detection).

¹⁸⁵ See, e.g., OTA Report, *supra* note 158, at 148-50 (giving various reasons for decompiling or disassembling programs). Concerning bug-fixing and adaptations, see, e.g., 17 U.S.C. sec. 117; Pamela Samuelson, *Modifying Copyrighted Software: Adjusting Copyright Doctrine to Accommodate a Technology*, 28 *Jurim. J.* 179 (1988). An example of reverse engineering to detect infringement is *E.F. Johnson Co. v. Uniden Corp. of Am.*, 623 F. Supp. 1485 (D. Minn. 1985). Reverse engineering of software for purposes such as those identified in the text may be less onerous than reverse engineering to make a directly competing non-identical clone because the reverse engineer may not have to analyze the whole program but only the parts where the “bug” is located or where necessary to add a particular feature.

¹⁸⁶ See *Sega Enterprises Ltd. v. Accolade, Inc.*, 977 F.2d 1510 (9th Cir. 1992); *Sony Computer Entertainment, Inc. v. Connectix Corp.*, 203 F.3d 596 (9th Cir. 2000). See also *Atari Games Corp. v. Nintendo of Am, Inc.*, 975 F.2d 832 (Fed. Cir. 1992)(reverse engineering to develop games that could be played on Nintendo console); *Vault Corp. v. Quaid Software Ltd.*, 847 F.2d 255 (5th Cir. 1988)(reverse engineering of copy-protection software to make spoofing software). Reverse engineering of software has sometimes been done to develop a complementary service. See, e.g., *Allen-Myland v. IBM Corp.*, 746 F.Supp. 520 (W.D. Pa. 1990)(engineering service reverse engineered IBM software to aid in reconfiguration of leased computers for subsequent lease customers); *Hubco Data Prods. Corp. v. Management Assistance, Inc.*, 219 U.S.P.Q. 450 (D. Idaho 1983)(reverse engineering to discover portions of MAI code that blocked use of advanced features to enable reverse engineer to provide service of providing MAI customers with cheaper way to access the advanced features). See also *Bateman v. Mneumonics, Inc.*, 79 F.3d 1532, 1545-47 (11th Cir. 1996)(compatibility considerations limit scope of copyright protection in programs).

¹⁸⁷ Other legal commentators have concluded that the economic consequences of reverse engineering in the software industry are benign. See, e.g., *Graham & Zerbe*, *supra* note 162; *Lemley & McGowan*, *supra* note 47. However, we believe that these analyses of the economic effects of reverse engineering were incomplete.

1. Incentives for Interoperable or Non-Interoperable Strategies

Before considering the role that reverse engineering plays in the interoperability debate, it is first important to grasp some basic concepts about the incentives of firms to design their systems to be interoperable or non-interoperable. A system, for these purposes, consists of two complementary pieces, such as a platform (e.g., the Sega Genesis machine or Microsoft's Windows operating system program) and applications designed to run on it (e.g., Sega's Sonic Hedgehog game or Lotus 1-2-3).¹⁸⁸ In the software industry, platforms and applications are not just complementary products; they are complementary parts of a system by virtue of their conformity to interfaces necessary to achieving interoperability. Platforms are typically designed first. If an applications developer wants to make a program that will fully interoperate with a particular platform, it must have access to very precise details about how the platform receives and sends information.¹⁸⁹ Collectively, these details are known as application programming interfaces (APIs). Some platform developers publish interfaces; some license them freely; and others maintain their APIs as closely held trade secrets.¹⁹⁰

The developer of a new platform might decide to publish its interfaces or make them available under open license terms—acts that make reverse engineering unnecessary—in order to make it easy for applications developers to adapt existing applications or make new applications for the platform. An important incentive to open interfaces is to drive demand for the new platform.¹⁹¹ Only if desirable applications are available for the platform will consumer demand for the platform skyrocket. In the 1980's, for example, IBM, then a new entrant into the personal computer (PC) market, published technical specifications for the PC and required Microsoft to broadly license the APIs to its operating system to enable applications developers to write programs for the IBM PC.¹⁹² This resulted in “[a] large library of off-the-shelf IBM PC compatible

¹⁸⁸ Applications programs can sometimes serve as platforms for applications that interoperate with them. See, e.g., *Lewis Galoob Toys, Inc. v. Nintendo of Am., Inc.*, 964 F.2d 965 (9th Cir. 1992), cert. denied, 507 U.S. 985 (1993)(game genie program designed to interoperate with Nintendo games and change certain aspects of game displays).

¹⁸⁹ One important role for platforms is to provide certain commonly needed services to applications. It is typically more efficient for the platform to do this, rather than requiring all developers of applications to write redundant code to do the same thing. However, applications developers need to know how to invoke needed platform functionality. This requires knowing how the platform expects to receive instructions from applications for that function to be successfully invoked and how it sends information pertinent to that functionality. See Band & Katoh, *supra* note 162, at 7.

¹⁹⁰ See, e.g., Cohen, *supra* note 162, at 1094; Manifesto, *supra* note 15, at 2402-03.

¹⁹¹ Another incentive to open interfaces is to aid development of an open source platform capable of supporting a range of applications. The Linux/GNU operating system is the most widely known open source platform. See, e.g., PETER WAYNER, *FREE FOR ALL: HOW LINUX AND THE FREE SOFTWARE MOVEMENT UNDERCUT THE HIGH TECH TITANS* (2000). The popularity of some open source platforms, such as the Apache web server, has caused commercial firms such as IBM Corp. to include it in IBM systems. *Id.* at 181-83.

¹⁹² “The IBM PC was the first deliberately open computer architecture, a fundamental insight that shaped the future of personal computing. From the very start, Boca Rotan [where IBM developed the PC] recognized that the best way to make the PC the industry standards was to publish all its technical specifications and make it easy for third parties to build add-on devices or write PC software applications, a principle that took Apple years to understand.” CHARLES H. FERGUSON & CHARLES R. MORRIS,

applications software (particularly Lotus 1-2-3) [that] made the IBM PC an attractive platform.”¹⁹³ This allowed the IBM PC to rapidly achieve a substantial market success.¹⁹⁴

Publishing or broadly licensing interfaces can, however, be risky for platform developers, even if beneficial for consumers and competitors. Hewlett-Packard and Dell are among the makers of IBM-compatible PCs who took advantage of IBM’s decision to embrace open architectures in the PC market. Consumers benefited from competition among IBM-compatible PCs and from a wide array of applications for this standard system. However, IBM lost market share in the PC market in part because the openness of its PC architecture enabled the PC to be “commoditized” or cloned.¹⁹⁵

Alternatively, firms may choose to keep their interfaces closed, not only as a defensive measure against the platform being commoditized, but as an offensive measure to capture the market.¹⁹⁶ Proprietary interfaces give the platform developer considerable control over applications available for the platform, in particular, the ability to insist that its own applications not be available for rival platforms.¹⁹⁷ The platform owner can ensure exclusivity either by developing the applications in-house, or by making exclusivity a condition of licensing. Firms that tried to keep their interfaces proprietary include Sega and Nintendo. Both forbade licensees from making games for other platforms, and both initiated lawsuits to stop unlicensed entrants, such as Accolade, from making games for their proprietary platforms or adapting games made for other platforms (recall that Accolade made games for IBM PCs).¹⁹⁸ The focus here is not on their attempts to stop software development for their platforms, but on their insistence that

COMPUTER WARS : HOW THE WEST CAN WIN IN A POST-IBM WORLD 52 (1993). Band and Katoh emphasize IBM’s insistence on requiring Microsoft to broad license APIs for its operating system for the PC. Band & Kato, *supra* note 162, at 30. A more recent example of a firm that freely publishes interface specifications for its platform is the maker of the popular Palm Pilot system. See, e.g., Douglas Lichtman, *Property Rights in Emerging Platform Technologies*, 29 J. Legal Stud. 615, 616 (2000).

¹⁹³ See, e.g., Band & Katoh, *supra* note 162, at 30.

¹⁹⁴ *Id.* (“in 1984 alone, IBM’s PC revenues were \$4 billion”).

¹⁹⁵ *Id.* at 31 (“By the early 1990’s, IBM sold only 23% of the IBM compatible PCs worldwide...”). Other reasons IBM had difficulty in controlling the PC market was that “in essence [it] ceded control of the microprocessor architecture to Intel and the operating system architecture to Microsoft.” *Id.* at 30. As IBM’s fortunes waned, Microsoft’s soared. From 1982 to 1993, “Microsoft’s annual revenues went from \$24 million to \$4.1 billion and its profits from \$3.5 million to \$794 million.” *Id.* at 31.

¹⁹⁶ See, e.g., Thomas Piraino, *Identifying Monopolists’ Illegal Conduct Under the Sherman Act*, 75 N.Y.U. L. Rev. 809, 888-89 (2000)(quoting a Microsoft manager’s internal email: “to control the APIs is to control the industry”). See also JERRY KAPLAN, *STARTUP: A SILICON VALLEY ADVENTURE* 49-50 (1994) (“our value is in the APIs” and “the real wars [in the computer industry] are over control of the APIs”).

¹⁹⁷ Since the platform developer knows its own APIs, it can easily supply them to applications programmers within the firm. Although the platform developer may also seek to attract independent applications developers to its platform, it may provide independent software vendors with less complete interface information and perhaps delayed access as compared with that provided within the firm. Microsoft’s practices in this regard were an important reason why the Dept. of Justice recommended breaking Microsoft into two firms, one an operating systems company and the other an applications development firm. Piraino recommends addressing this problem by ordering Microsoft to give applications programmers open access to Windows APIs. Piraino, *supra* note 196, at 888.

¹⁹⁸ *Sega*, 977 F.2d 1510 (9th Cir. 1992); *Atari Games*, 975 F.2d 832 (Fed. Cir. 1992).

such development occur under license. Licensing would allow them to impose exclusivity.

By keeping its interface proprietary, and providing an exclusive set of applications, a platform owner has some hope of exploiting “network effects”¹⁹⁹ to become a de facto standard in the market. In fact, a single “killer app” may suffice.²⁰⁰ The more successful a proprietary platform becomes, the easier it is to attract software developers, and the easier it is to attract consumers, both of which reinforce the system’s market dominance. At the same time, rivals may be forced out of the market, and entry deterred. If the dominant firm has a proprietary interface, an entrant faces the difficulty of entering at two levels: platform development and software development. Apple Computer and Sega are among the platform developers that hoped to achieve substantial market penetration with non-interoperable systems.

But just as publishing interfaces can be risky, so can the strategy of keeping them closed. If applications developers and consumers are not attracted to the system, losses can be considerable.²⁰¹ Even if initially successful, a non-interoperable system may lose out over time if other firms develop new systems to wrest away the incumbent’s market share. Sega, for example, was a second comer to the game system market, entering after Nintendo’s Entertainment System (NES) had achieved substantial market success.²⁰² Sega’s Genesis system offered some features the NES lacked, as well as certain new programs (notably one featuring a sonic hedgehog) that drew customers to the Genesis system. Later, Sega dropped out of the game system market, opting instead to develop games for other systems.²⁰³ The current market leader in the game system market is

¹⁹⁹ See, e.g., Michael L. Katz and Carl Shapiro, *Systems Competition and Network Effects*, 8 J. Econ. Persp. 93 (1994)(discussing network effects); Lemley & McGowan, supra note 47. Entrepreneur Jerry Kaplan offers this down-to-earth explanation of the phenomenon:

Creating an API is like trying to start a city on a tract of land you own. First you try to persuade applications programmers to come and build their businesses on it. This attracts users, who want to live there because of all the wonderful services and shops the programmers have built. This in turn causes more programmers to want to rent space for their businesses, to be near the customers. When this process gathers momentum, it’s impossible to stop.

Once your city is established, owning the API is like being king of the city. The king gets to make the rules: collecting tolls for entering the city, setting the taxes that the programmers and users have to pay, and taking first dibs on any prime locations (by keeping some APIs confidential for personal use).

Kaplan, supra note 196, at 50.

²⁰⁰ See, e.g., Band & Katoh, supra note 162, at 30 (emphasizing the importance of Lotus 1-2-3 as contributing to the success of the IBM PC).

²⁰¹ See, e.g., Kelly Zito, *New Path For Sega*, S.F. Chronicle, p. E1, Aug. 12, 2001. Sega recently exited the game system market due to \$420 million in losses in 2000 on the Dreamcast system it introduced into the market in 1999. Sega’s new system met with resistance from applications developers who decided not to tailor games for it. See James Surowiecki, *Games People Play*, New Yorker, May 7, 2001, at 36.

²⁰² See, e.g., DAVID SHEFF, *GAME OVER: HOW NINTENDO ZAPPED AN AMERICAN INDUSTRY, CAPTURED YOUR DOLLARS, AND ENSLAVED YOUR CHILDREN* 352-53 (1993)(discussing Sega’s entry into the game system market and its effort to gain market share against Nintendo’s Entertainment System).

²⁰³ See, e.g., Zito, supra note 201, at E1. Sega will now concentrate on the sale of games for other platforms because this is a more profitable line of business. Id. at E4.

Sony's PlayStation,²⁰⁴ whose lead is about to be challenged by new entrant Microsoft's Xbox system.²⁰⁵ In the game system market, platform developers typically lose money on sales of consoles, making up losses on sales of games and peripherals.²⁰⁶

In contrast to the game system market, which has been characterized by serial monopolies, Microsoft's operating system program has become a de facto standard platform for applications running on personal computers, a monopoly that has been durable over many years.²⁰⁷ Over this period, Microsoft's operating systems interfaces have become more complex, and its licensing practices as to interfaces more restrictive. One explanation for the increasing complexity of Windows interfaces is that Microsoft has responded to some innovative applications by integrating them into the Windows operating system program (by a strategy sometimes known as "embrace and extend").²⁰⁸ This has undermined the market of some competing applications, such as Netscape's browser, and threatened their viability. Microsoft has also responded to competition in the applications market by providing suites of popular applications (e.g., Microsoft Office) at attractive prices so that consumers will buy the suites instead of separate products from competing vendors. In addition, Microsoft has responded aggressively to innovations with potential to become alternative platforms to Windows, such as the Java programming system.²⁰⁹ Even if much is disputed about Microsoft's conduct in preserving its operating systems monopoly, no one would dispute that Microsoft's control

²⁰⁴ Id. Sony has an installed base of 85 million PlayStations. Id.

²⁰⁵ See, e.g., Chris Gaither, Microsoft Delays Release of Xbox Game System by a Week, *New York Times*, p. C15, Sept. 22, 2001. Microsoft hopes to ship 1.5 million consoles by the end of the 2001 holiday season. Id.

²⁰⁶ Id. See also Surowiecki, *supra* note 201, at 36 ("Sony loses money on every PlayStation 2 it makes"). Game consoles are expensive because of the many hardware components (semiconductor chips, graphics cards, memory, and the like). Id. Consumers are sufficiently sensitive to the costs of the consoles that it makes commercial sense to take losses on sales of consoles that can then be made up on sales of applications. A large installed base is helpful to achieving this objective. See, e.g., William E. Cohen, *Competition and Foreclosure in the Context of Installed Base and Compatibility Effects*, 64 *Antitrust L.J.* 535 (1996).

²⁰⁷ *U.S. v. Microsoft Corp.*, 243 F.3d 34, 54-58 (D.C. Cir. 2001)(finding Microsoft had monopoly power in the market for operating systems for Intel-compatible PCs). See also Franklin M. Fisher & Daniel L. Rubinfeld, *U.S. v. Microsoft: An Economic Analysis*, *Antitrust Bull.* 1, 13-19 (Spring 2001) (discussing Microsoft's monopoly). It is worth pointing out that operating systems with open interfaces can be supplied by competing firms. See, e.g., Wayner, *supra* note 191, at 41-52 (discussing the successful struggle to open the Unix operating system).

²⁰⁸ The Dept. of Justice charged that Microsoft's decision to integrate its Internet Explorer browser into the Windows operating systems was intended to harm the market for Netscape's competing browser. See, e.g., *U.S. v. Microsoft*, 243 F.3d 34, 84-97 (D.C. Cir. 2001)(discussing theory but remanding case to trial court for further findings). See also John Heilemann, *The Truth, The Whole Truth and Nothing But the Truth*, <http://www.wired.com/wired/archives/8.11/microsoft.html>.

²⁰⁹ The World Wide Web opened up new opportunities for evolution of new platforms, such as browser software, for which applications could be written. See, e.g., Fisher & Rubinfeld, *supra* note 208, at 20-23. See also Mark A. Lemley & David McGowan, *Could Java Change Everything? The Competitive Propriety of a Proprietary Standard*, 43 *Antitrust Bull.* 715 (1998).

over the APIs for developing applications for the Windows platform is an important source of its enduring power in this market.²¹⁰

Into this strategic environment, we now introduce reverse engineering. Platform developers typically copyright operating system programs; they may also patent some components of their systems; but APIs are typically maintained as trade secrets.²¹¹ If reverse engineering is unlawful or if the platform is otherwise immune from reverse engineering (e.g., because the interfaces are too complicated or change rapidly),²¹² trade secrets can be a very effective form of intellectual property protection for platform APIs.²¹³ If reverse engineering is both lawful and feasible, trade secrecy protection for platform APIs is at risk. Reverse engineering clearly threatens to upset a platform developer's non-interoperability strategy, whether unlicensed entry occurs at the applications level or at the platform level. From the standpoint of an unlicensed applications developer, reverse engineering offers a means of achieving compatibility between its products and the large installed base of a successful system.²¹⁴ Although it would have been easier and quicker to license the Sega Genesis interface, Accolade would have had to stop writing for other platforms, due to Sega's insistence on exclusivity.²¹⁵ Reverse engineering gave Accolade an alternative way to access the Sega interfaces and enter the market with competing applications.

2. Welfare Effects of Reverse Engineering To Achieve Interoperability

Table 3 compares the principal economic effects of allowing or disallowing reverse engineering to achieve interoperability in the software industry. Although we use similar criteria as for traditional manufacturing and semiconductor chips,²¹⁶ the welfare

²¹⁰ See supra note 196 (quoting a Microsoft manager on the importance of APIs). See also U.S. v. Microsoft, 243 F.3d 34, 55-56 (D.C. Cir. 2001) (discussing the applications barrier to entry that protects a dominant operating system irrespective of quality).

²¹¹ Courts have held that copyright protection does not extend to interfaces of computer programs. See, e.g., Computer Associates Int'l v. Altai, Inc., 983 F.2d 693 (2d Cir. 1992). Patents may sometimes protect aspects of program interfaces. See, e.g., Atari Games Corp. v. Nintendo of Am., Inc., 30 U.S.P.Q. 2d (BNA) 1420 (N.D. Cal. 1993) (granting partial summary judgment to Nintendo on patent infringement claims as to interface components).

²¹² See infra Section VI-B(2).

²¹³ Economists Joseph Farrell and Michael Katz consider intellectual property as determining whether a rival network will be compatible. They do not distinguish platforms from applications, but argue that intellectual property in the interface increases the incentive for quality improvements in a system as a whole. See Joseph Farrell & Michael L. Katz, *The Effects of Antitrust and Intellectual Property Law on Compatibility and Innovation*, 43 Antitrust Bull. 609 (1998). We caution, however, that intellectual property in the interface may be unnecessary if platforms and applications are themselves protected. With intellectual property in platforms and applications, intellectual property in the interfaces may serve no beneficial purpose and only allow developers to leverage market power in a way that was unintended as a matter of intellectual property law.

²¹⁴ The unlicensed entrant who reverse engineers the APIs and then sells system components may benefit by substantial expenditures made by the platform provider to promote the platform in the market. Microsoft's Xbox system will be launched with a \$500 million marketing campaign. Gaither, supra note 205, at C15.

²¹⁵ *Sega*, 977 F.2d at 1514.

²¹⁶ The price and wasted costs criteria are identical to the earlier charts, although price is now a more complicated phenomenon because we must consider the effects of pricing of both platform and

effects of reverse engineering rules in the software industry are more complicated and ambiguous. We explain the reasons for this below.

Table 3
Social Calculus of Reverse Engineering of Software for Purposes of Interoperability

Social welfare criterion	RE legal	RE illegal
Incentives to develop platform	lower (but adequate?)	higher (too high?)
Incentives to develop applications	high(er?)	high
System Price		
short run	?	?
long run (tipping)	lower	higher
Duplicated costs	lower?	higher?

The conclusion about which we have the greatest confidence is that incentives to invest in platform development will be lower if reverse engineering is lawful. If third parties can legally reverse engineer program interfaces, this erodes the market power of a non-interoperable platform developer.²¹⁷ In this respect, reverse engineering poses the same threat in the software industry as in traditional manufacturing industries: it erodes market power by facilitating unlicensed entry or by inducing licensing on terms more favorable to the licensee than if reverse engineering was prohibited.²¹⁸ Of course, this does not necessarily mean that reverse engineering should be made illegal in order to protect platform developers. This depends on the cost and time required for reverse engineering. Because decompilation and disassembly are time-consuming and resource-intensive, these forms of reverse engineering do not, we believe, significantly undermine incentives to invest in platforms.²¹⁹

applications. Incentives to innovate must, however, be split into two components, one focusing on incentives to develop platforms and one focusing on incentives to develop applications. Because platforms are typically developed before applications, applications are an important category of follow-on innovation. We could have broken incentives for follow-on innovation down further into incentives to improve platforms and then incentives to improve applications, but this would needlessly complicate the main points we seek to make in this subsection about systems competition issues.

²¹⁷ Graham and Zerbe emphasize this factor in their economic analysis of reverse engineering in the software industry. See Graham & Zerbe, *supra* note 162, at 122.

²¹⁸ See *supra* Section II-B.

²¹⁹ It may be worth noting that reverse engineering in the software industry rarely involves development of a competing platform, but more often involves entry at the applications level. In *Sony v. Connectix*, discussed *supra* note 172 and accompanying text, the platform developer was actually losing money on the sale of each platform. See *supra* note 206 and accompanying text. One might have expected Sony to welcome new entrants to expand its installed base without causing the firm additional losses, but this was not Sony's response. Sony complained of reputational damage to its system because PlayStation games operated less well on the emulator platforms. See, e.g., *Connectix*, 203 F.3d at 608-09.

As for applications, there are strong incentives to develop them whether interfaces are open or closed. If interfaces can lawfully be reverse engineered and hence are potentially open, any software developer will be able to develop applications for the platform, not just the developers licensed by the platform developer. Accolade, for example, adapted its Mike Ditka football game program to run on the Sega Genesis system, increasing the number of applications available for that platform. As this example shows, open interfaces not only facilitate third party development of applications, but also the adaptation of applications to multiple platforms, which saves software development costs.²²⁰

However, there are also strong incentives to develop applications when interfaces are proprietary and cannot be reverse engineered. The developer of a non-interoperable platform wants a large installed base of customers. It can attract customers by providing a large number of attractive applications, especially those that may be exclusive to that platform.²²¹ Independent software developers may easily be drawn to developing applications if they think the platform will emerge as the dominant one. If the platform is struggling to gain a toehold, it may have an even larger incentive to develop applications, perhaps doing so in-house or subsidizing independent developers who might otherwise be reluctant.²²²

²²⁰ Church and Gandal address the question of software development under open and closed interfaces, although they do not assume that independent software vendors write software for all platforms simultaneously, even when interfaces are open. See Jeffrey Church & Neil Gandal, *Integration, Complementary Products, and Variety*, 1 J. Econ. & Mgmt. Strategy 25 (1992). There is no opportunity in their model to avoid software development costs by making each application compatible with all platforms, even when interfaces are open. Church and Gandal argue that despite the social benefits of open interfaces, firms have an incentive to choose proprietary interfaces. It seems that changing the model such that software vendors can write for all platforms simultaneously under a system of open interfaces would reinforce the conclusion that the firms' incentives to "go proprietary" are detrimental not only to the firms, but to consumers.

²²¹ Of course, incentives to develop applications also depend on the extent of intellectual property protection available to them. If such protection is weak and competitors can imitate design elements of a proprietary application, this may erode the market advantage the platform owner had hoped to garner through its investment. This helps to explain the "look and feel" lawsuits of the late 1980's and early 1990's. See, e.g., *Apple Computer Inc. v. Microsoft Corp.*, 35 F.3d 1435 (9th Cir. 1994)(rejecting Apple's claim that the look and feel of Microsoft's graphic user interface (GUI) infringed Apple's copyright in the Macintosh GUI); *Lotus Development Corp. v. Borland Int'l*, 49 F.3d 807 (1st Cir. 1995)(rejecting Lotus' claim that the emulation interface of Borland's Quattro Pro spreadsheet program infringed Lotus 1-2-3). See also *Data East USA, Inc. v. 204* (9th Cir. 1988)(no infringement where similarities between two independently developed karate programs lay in standard features to be expected of such games).

²²² The platform developer's ability to attract developers to develop applications for the platform and to recoup subsidies incurred to attract applications developers may be negatively affected to some degree by a rule favoring reverse engineering. If reverse engineering is lawful, licensed developers may worry about their recoupment of R&D expenses if unlicensed entrants can now offer competing applications for the platform—and can do so without paying royalties to the platform developer for the right to make applications for the platform. However, there are counterbalancing factors: first, licensed applications developers will have significant first-mover advantages in the applications market as compared with reverse engineers because decompilation and disassembly are so difficult and time-consuming, and second, over time, licensed independent software vendors may be in a better position to negotiate with platform developers for terms more favorable to them if reverse engineering is a legal option. Especially if the applications developer has had a "hit" in the applications market for a non-interoperable system, it may be

Incentives to develop platforms and applications are naturally tied up with equilibrium prices. Two key market ingredients that affect pricing are: 1) whether systems are compatible or incompatible, and 2) whether platform owners supply their own applications. We shall refer to the latter as “integrated” systems, in contrast with “unintegrated” systems in which independent firms supply applications for separately owned platforms. We think the most natural stylization of the pricing problem is that closed interfaces lead to incompatible and integrated systems,²²³ while open interfaces lead to compatible and unintegrated systems.

We have not found an economic model with which to compare prices or incentives to develop platforms and applications in these two market structures. The economics literature has mainly compared two types of integrated ownership, namely, with interoperable and noninteroperable applications. That literature yields inconclusive results.²²⁴ In any case, it seems that integrated ownership of compatible systems would likely be unstable. With open interfaces, achieved by reverse engineering or otherwise, independent applications developers will enter with compatible applications, and platform providers will enter with compatible platforms. Both undermine the integrated market structure.

able to negotiate more favorable terms, such as a right to develop its applications for more than one platform. See, e.g., Surowiecki, *supra* note 201, at 36.

²²³ Platform owners with closed interfaces may contract with independent software vendors for development of applications, but we assume that they do so on license terms that capture much of the value for the platform owner, and under terms of exclusivity. Thus, prices should not depend very sensitively on whether platform owners contract with independent software vendors or develop their applications in-house. For our purposes, the key distinction is whether systems are integrated and incompatible, or unintegrated and compatible.

²²⁴ See, e.g., Carmen Matutes and Pierre Regibeau, *Mix and Match: Product Compatibility Without Network Externalities*, 19 RAND J. Econ. 221 (1988). These authors argue that, with two firms, and demand conditions such that each consumer uses only one application, systems prices will be higher when the integrated systems are incompatible than when compatible. The same result recurs in a different model by Joseph Farrell, Hunter Monroe, and Garth Saloner, *The Vertical Organization of Industry: Systems Competition versus Component Competition* 7 J. Econ. and Mgmt Strategy 143 (Summer 1988), but the latter also show that with more than two firms, the result on prices can be reversed; systems prices can be higher when systems are incompatible. In a model with two systems, Jeffrey Church and Neil Gandal, *supra* note 220, conclude that incompatibility leads to lower systems prices than an unintegrated system.

An intuition for higher prices with compatible, but integrated, systems is that platform owners will compete less fiercely because a seller’s loss in platform sales can be mitigated by increased sales of his application to purchasers of the other platform. A second intuition follows the observations of Augustin Cournot, *Researches into the Mathematical Principles of the Theory of Wealth* (1838) (in French, English translation by Nathaniel Bacon, published by Oxford Press in 1927), who observed that if a single firm sells complementary pieces of a whole, it will do so at a lower total price than two firms selling the components separately. The total price offered by the integrated firm will also yield more profit than the (higher) joint price charged by separate firms. Some commentators have relied on Cournot’s insights to argue that consumers would be better off if platform developers controlled the applications market through licensing of proprietary interfaces. See, e.g., Lichtman, *supra* note 192. We question the applicability of Cournot’s analysis to software system markets, as these markets have more complex dynamics than the fixed industrial complements on which Cournot focused.

It is difficult to compare prices between the two market structures. In an integrated system, platforms and applications may be sold as a unit, but may also be sold separately with cross subsidies between system components. In an unintegrated system, platforms and applications are priced and sold separately, at prices that are governed by the degree of competition in both markets, and possibly by intellectual property law. Our entries in Table 3 are inconclusive about pricing, but indicate that when reverse engineering is illegal, so that systems may be integrated and incompatible, prices may be higher in the long run than in the short run due to the threat of tipping.

“Tipping” means that a single interface succeeds in becoming the standard in the market, and hence a monopoly. Such tipping would be detrimental to consumers, but beneficial to the winning platform owner. By buying up talented independent applications developers, entering into exclusive licensing agreements with them, or simply attracting them due to its large installed base, a platform owner may create sufficient network externalities to drive out rivals, and remain the sole platform provider.²²⁵

A right to reverse engineer may neutralize this threat of “tipping.” If the interface becomes open through reverse engineering or otherwise, other firms can develop platforms to compete with the proprietary platform, and thereby undermine the latter’s monopoly pricing strategy. Insofar as this interface becomes a de facto standard, consumers will benefit because more applications will be available for the platform and applications developers will be in a better position to negotiate with firms competing in the platform market for better access to interface information.

Wasted costs is the fourth social welfare criterion. It too yields somewhat mixed policy prescriptions. Duplicated or wasted costs may arise in the software industry from at least three activities: (1) in the act of reverse engineering itself (costs wasted by the reverse engineer); (2) in devising ways (e.g., technical protection measures) to make interfaces difficult or impossible to reverse engineer (costs wasted by the platform developer);²²⁶ and (3) in developing different applications for different interfaces rather than the same applications for all interfaces (costs wasted by applications developers generally). A prohibition on reverse engineering would avoid the first two, but may well encourage the third. A platform provider can, of course, avoid the first cost by licensing, and as in other industrial contexts, a legal rule in favor of reverse engineering may provide powerful incentives for firms to license to avoid having their products reverse engineered.

²²⁵ Several commentators have argued against intellectual property in interfaces on these grounds. See Jeffrey Church and Roger Ware, *Network Industries, Intellectual Property Rights and Competition Policy*, in *COMPETITION POLICY AND INTELLECTUAL PROPERTY RIGHTS IN THE KNOWLEDGE-BASED ECONOMY* (Robert Anderson and Nancy T. Gallini, eds., 1998) and Lemley and McGowan, *supra* note 47, at 525. But Farrell and Katz, *supra* note 213, argue that intellectual property in interfaces can give firms incentives to improve their platforms. We caution, however, that if platforms and applications are themselves protected by appropriate intellectual property, then providing intellectual property on interfaces might only give platform owners a means to leverage their market power beyond that which was intended by Congress.

²²⁶ See, e.g., Cohen, *supra* note 162, at 1094. See also *infra* Section VI-B(2) for a discussion of the policy implications of efforts to thwart reverse engineering by making one’s product difficult to reverse engineer.

It is difficult to integrate these disparate welfare effects into an unassailable view as to whether reverse engineering for interoperability purposes should be legal. On balance, we believe that consumers benefit from interoperability because it encourages the development of a larger variety of software applications from a wider array of software developers with less wasted applications development costs. Incentives to develop platforms are generally adequate owing to the high costs and difficulties of reverse engineering software. Furthermore, interoperability lessens the potential for tipping into monopoly. Reverse engineering to achieve interoperability may also lessen a monopoly platform provider's market power by providing applications developers with an alternative means of entry if the monopolist's licensing terms are unacceptable.

C. Reverse Engineering of Software And Contract Law

Another strategy for prohibiting decompilation and other forms of reverse engineering of programs has been through contractual restrictions, often by licenses inserted in boxes of packaged software.²²⁷ The enforceability of such restrictions has been a highly contentious legal issue both in the U.S. and abroad.²²⁸ The caselaw in the U.S. is in conflict on the enforceability of anti-reverse engineering clauses in software contracts.²²⁹ Irresolution in the caselaw might suggest the need for a legislative

²²⁷ See, e.g., Mark A. Lemley, *Beyond Preemption: The Law and Policy of Intellectual Property Licensing*, 87 Calif. L. Rev. 111, 129 (1999).

²²⁸ The European Union has declared that anti-decompilation clauses in software contracts are null and void. See European Software Directive, *supra* note 177, Art. 9(1). The principal reason EU chose to make anti-decompilation clauses unenforceable was to create incentives for firms to license interface information on a reasonable basis so that the second comers would not resort to reverse engineering to get this information. See Official Commentary, reproduced in Czarnota & Hart, *supra* note 177, at 76-80. A few other countries, notably Australia, have followed suit. See, e.g., Jonathan Band, *Software Reverse Engineering Amendments in Singapore and Australia*, J. Internet L. 17, 20 (Jan. 2000).

²²⁹ Courts have sometimes rejected reverse engineering defenses in trade secrecy cases because this activity exceeded the scope of licensed uses of the software. See, e.g., *Technicon Data Systems Corp. v. Curtis 1000, Inc.*, 224 U.S.P.Q. 286 (Del. Ch. 1984) (consultant to hospital used improper means to obtain trade secret interface information by wiretapping the hospital's licensed software system to study the manner in which the server software exchanged data with the client software because it had not been authorized by the hospital, and even if it had been, the action would have breached restrictive terms in the license). See also *DSC Communications Corp. v. Pulse Communications, Inc.*, 170 F.3d 1354 (Fed. Cir. 1999) (triable issue of fact as to whether Pulsecom's use of "snooper board" at telephone company to get access to interface information about DSC's software was misappropriation of trade secret in view of restrictions in the telephone company's license to use DSC's software). For a non-software case in which an anti-reverse engineering clause was enforced, see *K&G Oil & Tool Service Co. v. G&G Fishing Tool Service*, 158 Tex. 594, 314 S.W.2d 782 (1958).

In some cases, courts have declined to enforce shrinkwrap license restrictions against reverse engineering, sometimes because of a conflict between the clause and federal intellectual property policy. The principal case is *Vault Corp. v. Quaid Software Ltd.*, 847 F.2d 255 (5th Cir. 1988) in which the maker of a copy-protection program sought to enforce an anti-reverse engineering clause of a shrinkwrap license under Louisiana law against a firm that had reverse engineered the copy-protection scheme. The Court of Appeals held that: "[t]he provision in Louisiana's License Act, which permits a software producer to prohibit the adaptation of its licensed computer program by decompilation or disassembly, conflicts with the rights of computer program owners under [the copyright law] and clearly 'touches upon an area' of federal copyright law. For this reason...we hold that this provision of Louisiana's License Act is preempted by federal law, and thus that the restriction in Vault's license agreement against decompilation

resolution. However, legislative approaches have also been contentious, as witnessed by the controversy over the model law now known as the Uniform Computer Information Transactions Act (UCITA).²³⁰

UCITA aims to resolve the decades' long controversy about shrinkwrap and other mass market licenses for software.²³¹ As long as a user has had a reasonable opportunity to review the terms of a license, merely using the software may constitute the user's assent to the license terms.²³² Endorsing freedom of contract as a core value,²³³ UCITA generally presumes license terms to be enforceable unless unconscionable.²³⁴ Yet, owing to lingering concerns about imbalance in UCITA,²³⁵ this model law now provides that if "a term of a contract violates a fundamental public policy, the court may enforce the remainder of the contract without the impermissible term, or so limit the application of the impermissible term as to avoid any result contrary to public...."²³⁶ UCITA also recognizes that if federal law preempts one of its provisions, that provision is "unenforceable to the extent of the preemption."²³⁷

or disassembly is unenforceable." *Id.* at 270. See also *Symantec Corp. v. McAfee Associates*, 1998 WL 740798 (N.D. Cal. 1998) (state unfair business practice claim based on reverse engineering of another firm's program in violation of license agreement held preempted by copyright law).

Some cases have also ruled against enforcing shrinkwrap licenses as a matter of contract law, either as contracts of adhesion or as lacking mutuality of consent, although the caselaw is mixed on this issue as well. Compare *Step Saver Data Systems v. Wyse Technology*, 939 F.2d 91 (3d Cir. 1991) (shrinkwrap license not enforceable as a matter of contract law) and *ProCD Inc. v. Zeidenberg*, 86 F.3d 1447 (7th Cir. 1996) (enforcing shrinkwrap license restriction). See also Band & Katoh, *supra* note 151, at 221; L. RAY PATTERSON & STANLEY W. LINDBERG, *THE NATURE OF COPYRIGHT—A LAW OF USER'S RIGHTS* 220 (1991).

²³⁰ Uniform Computer Information Transactions Act, available at <http://www.ucitaonline.com/ucita.html>. With some consumer protection modifications, UCITA was enacted and is in force in Maryland. The Virginia also enacted it with a two year moratorium. A status report on state enactments of UCITA is available at <http://www.ucitaonline.com/slhpsus.html>.

²³¹ See, e.g., Robert W. Gomulkiewicz, *The License Is the Product: Comments on the Promise of Article 2B for Software and Information Licensing*, 13 *Berkeley Tech. L.J.* 891 (1998) (reviewing history of model law project and issues).

²³² See UCITA, *supra* note 230, secs. 112, 210-11.

²³³ See Reporter's Notes to UCITA sec. 104 (UCITA conforms to "fundamental policy of the United States which holds that freedom of contract governs").

²³⁴ UCITA, *supra* note 230, sec. 111. UCITA does limit licensor freedom to some degree, for example, as to choice of law clauses in consumer contracts. *Id.*, sec. 109. To the extent UCITA might conflict with an applicable consumer protection law, the latter will govern. *Id.*, sec. 105(c). Some commentators have pointed out that most consumer protection laws apply to sales of goods and not to licenses of goods, and hence sec. 105 may supply less protection to consumers than might be apparent. See, e.g., Jean Braucher, Memorandum, *The Uniform Computer Information Transactions Act (UCITA): Objections From The Consumer Perspective*, Aug. 15, 2000 (on file with the authors).

²³⁵ See, e.g., Charles R. McManis, *The Privatization (or "Shrinkwrapping") of American Copyright Law*, 87 *Calif. L. Rev.* 173, 187-90 (1999) (discussing compromise).

²³⁶ UCITA, *supra* note 230, sec. 105(b).

²³⁷ *Id.*, sec. 105(a).

The implications of these UCITA provisions for anti-reverse engineering clauses have been the subject of considerable debate.²³⁸ Some commentators believe that anti-reverse engineering clauses in mass market licenses should be unenforceable on copyright preemption grounds.²³⁹ Others have asserted that such clauses should be considered a misuse of intellectual property rights.²⁴⁰ Still others have suggested enforcing such license terms in negotiated licenses, but not in non-negotiated standard form contracts.²⁴¹ Another suggestion is to enforce them unless the firm imposing the license term has monopoly power.²⁴² A new doctrine of public interest unconscionability has also been proposed under which anti-reverse engineering clauses in mass market licenses would be unenforceable.²⁴³

Counterarguments abound as well.²⁴⁴ Critics point out that copyright preemption of contract terms is rare.²⁴⁵ Misuse of intellectual property rights is a doctrine of uncertain scope and application, and some have opined that it should extend no farther than antitrust law would.²⁴⁶ Because most consumers do not want to reverse engineer the software they buy, it may be difficult to challenge anti-reverse engineering clauses on unconscionability grounds.²⁴⁷ While antitrust and competition law may regulate anti-reverse engineering clauses in an appropriate case or context, no such claim has as yet been brought, let alone sustained.

²³⁸ See, e.g., Lemley, *supra* note 27; David McGowan, *Free Contracting, Fair Competition, and Article 2B: Some Reflections on Federal Competition Policy, Information Transactions, and "Aggressive Neutrality,"* 13 Berkeley Tech. L.J. 1173 (1998); Reporter's Notes to UCITA sec. 105.

²³⁹ See, e.g., McManis, *supra* note 235; David A. Rice, *Public Goods, Private Contract and Public Policy: Federal Preemption of Software License Prohibitions Against Reverse Engineering*, 53 U. Pitt. L. Rev. 543 (1992).

²⁴⁰ See, e.g., Lemley, *supra* note 227, at 151-58. But see Marshall Leaffer, *Engineering Competitive Policy and Copyright Misuse*, 19 U. Dayton L. Rev. 1087, 1106 (1994).

²⁴¹ See, e.g., David Nimmer, Elliot Brown, & Gary N. Frischling, *The Metamorphosis of Contract Into Expand*, 87 Calif. L. Rev. 17, 68 (1999). The Reporter's Notes to sec. 105(b) opined that anti-reverse engineering terms would likely be enforced as to negotiated contracts, but acknowledged as an open question whether they would be enforced in mass market licenses. See, e.g., McGowan, *supra* note 238, at 1195-98 (reviewing various iterations of the Reporter's Notes).

²⁴² See, e.g., Maureen O'Rourke, *Drawing the Boundary Between Copyright and Contract: Copyright Preemption of License Terms*, 45 Duke L.J. 479, 551 (1995). See also McGowan, *supra* note 237, at 1176-77 (raising question about enforcing such terms in concentrated markets).

²⁴³ See, e.g., J.H. Reichman & Jonathan A. Franklin, *Privately Legislated Intellectual Property Rights: Reconciling Freedom of Contract With Public Good Uses of Information*, 147 U. Pa. L. Rev. 875, 939 (1999).

²⁴⁴ See, e.g., Raymond T. Nimmer, *Breaking Barriers: The Relation Between Contract and Intellectual Property Law*, 13 Berkeley Tech. L.J. 827, 861-88 (1998) (responding to arguments based on preemption, misuse, and other doctrines).

²⁴⁵ See, e.g., Reporter's Notes to UCITA sec. 105(a). See also Lemley, *supra* note 227, at 144-50 (discussing limits of preemption doctrine as applied to licensing). In general, state contract claims are different enough in kind from copyright claims as to be beyond preemption. See, e.g., *National Car Rental System, Inc. v. Computer Associates Int'l, Inc.*, 991 F.2d 426 (8th Cir. 1993); *ProCD Inc. v. Zeidenberg*, 86 F.3d 1447 (7th Cir. 1996). But see Nimmer et al., *supra* note 241, at 42-57 (critical of preemption analysis in *ProCD*); Niva Elkin-Koren, *Copyright Policy and the Limits of Freedom of Contract*, 12 Berkeley Tech. L.J. 93 (1997).

²⁴⁶ See, e.g., Lemley, *supra* note 227, at 152, n. 188.

²⁴⁷ See, e.g., McGowan, *supra* note 238, at 1204-14.

Some legal commentators have pointed to collective action problems and negative externalities as impediments to achieving appropriate market outcomes via contract law that UCITA's freedom of contract policy assumes.²⁴⁸ In respect of anti-reverse engineering clauses in software licenses, Professor McGowan points out:

On average, consumers would probably assent to limitations relating to reverse engineering, their assent would be rational, and requiring evidence of deliberative assent therefore would increase transactions costs without yielding corresponding benefits that are relevant to federal policy concerns.... The collective product of such atomistic acts of assent, however, would pose the same risks for social welfare that advocates of legal rules facilitating reverse engineering... would like to ameliorate—lethargic transition among standard products and diminished production of works building upon ideas embedded in object code.²⁴⁹

There is a wider public interest in the availability of competitive products in the future that might be thwarted if anti-reverse engineering clauses were enforced. Third party effects of enforcing anti-reverse engineering clauses might, therefore, be harmful to consumer welfare. McGowan concludes that if “reverse engineering furthers copyright’s goal of promoting the dissemination and improvement of intellectual property [and] reverse engineering does not deprive authors of returns necessary to induce investment..., then competition policy would favor reverse engineering as a device to lower the cost of transition among standard products (thereby increasing allocative efficiency) without infringing on copyright goals or methodology.”²⁵⁰

As explained above, we believe that the welfare effects of reverse engineering in the software industry context are somewhat more complex than this. However, on balance, reverse engineering and interoperability are important because they likely promote development of a wider range of software from a broader array of developers than a market in which platform developers are insulated from reverse engineering. To the extent that enforcement of anti-reverse engineering clauses would have a detrimental effect on competitive development and innovation, legal decisionmakers may be justified in not enforcing them.²⁵¹

V. Reverse Engineering of Technically Protected Digital Content

The market for copyrighted works would seem to be in a transitional period. For many years, copyright industries have derived the bulk of their revenues from the sale of physical products, such as books and videocassettes, in the mass market. Advances in

²⁴⁸ See, e.g., Julie E. Cohen, Lochner in Cyberspace: *The New Economic Orthodoxy of “Rights Management,”* 97 Mich. L. Rev. 462, 536-38, 547 (1998).

²⁴⁹ McGowan, supra note 238, at 1213-14.

²⁵⁰ Id. at 1205-06.

²⁵¹ We agree with other commentators that the argument for non-enforcement of anti-reverse engineering clauses is strongest as to mass-market software and weakest as to negotiated agreements between sophisticated firms. See, e.g., O’Rourke, supra note 242; Reichman & Franklin, supra note 243. See also infra Section VI-B(1).

digital technology have opened up the possibility of a future in which a substantial portion of copyright industry revenues may come from mass-marketing of technically protected digital content.²⁵² Copyright industry groups persuaded Congress to provide legal reinforcements to these technical protections so that it would become illegal to circumvent technical measures used by copyright industries to protect their works and to develop or distribute circumvention technologies, hence, the Digital Millennium Copyright Act (DMCA) of 1998.²⁵³

Although the DMCA rules are not explicitly cast as restrictions on reverse engineering, that is their essential nature. Just as it is impossible to reverse engineer object code without decompiling or disassembling it, it is impossible to reverse engineer a technical protection measure without circumventing it. Someone who reverse engineers a technical protection measure will also generally need a tool in order to perform such reverse engineering activities, so by outlawing the making of circumvention technologies, the law indirectly restricts reverse engineering.

The DMCA's restrictions on reverse engineering represent an inversion of the rules that apply in other industrial contexts. Under the DMCA, reverse engineering of technical measures may be illegal except when authorized by a specific statutory or rule-making exception.²⁵⁴ Even when allowed, the DMCA strictly regulates what can be done with the resulting information.²⁵⁵ Even tools for reverse engineering are, for the most part, banned.²⁵⁶ The range of these restrictions is unprecedented in American law.

Section A will provide an overview of a future market in technically protected digital works that copyright industries envision. Section B will discuss the law pertaining to circumvention and circumvention tools in the pre-DMCA era. It will go on to consider the circumstances leading up to the DMCA and the complex architecture of the DMCA rules. Section C will explore the economics of the DMCA rules. It will explain why those rules are overbroad and how the rules might be reformed to be more economically sound.

A. Emerging Markets in Technically Protected Works

The idea of technically protecting digital forms of copyrighted works is not a wholly new one. In the 1980's some computer software developers used copy-protection technologies when mass-marketing their products. Two factors led to the abandonment of copy-protection measures for software: first, copy-protection measures were

²⁵² For a discussion of technical protection measures for digital content generally, see COMPUTER SCIENCE & TELECOMMUNICATIONS BOARD, NATIONAL RESEARCH COUNCIL, THE DIGITAL DILEMMA: INTELLECTUAL PROPERTY IN THE INFORMATION AGE 153-76 (2000) (cited hereinafter as "Digital Dilemma").

²⁵³ Pub. L. No. 105-304, 112 Stat. 2860 (1998). The anti-circumvention provisions of the DMCA have been codified at 17 U.S.C. secs. 1201-04.

²⁵⁴ See *infra* notes 287-94 and accompanying text for a description of exceptions, now codified at 17 U.S.C. sec. 1201(e)-(j).

²⁵⁵ These restrictions are discussed *infra* Section VI-A(5).

²⁵⁶ 17 U.S.C. sec. 1201(a)(2), (b)(1).

displeasing to major customers because they interfered with some legitimate uses of software products, such as making backup copies, and second, makers of some competing software decided to make their products available without copy-protection to give them a competitive advantage.²⁵⁷ This strategy worked well enough that copy-protection schemes for mass-marketed software died out in the marketplace. In the early 1990's digital audio tape (DAT) machines began to be sold into the consumer market with a built-in technical protection measure. The Audio Home Recording Act (AHRA) required that all consumer-grade DAT machines include a serial copy management system (SCMS) chip that allowed users to make individual personal use copies of DAT sound recordings, but SCMS ensured that these copies were programmed to degrade in quality if the users tried to make further copies of the copies.²⁵⁸ DAT technologies met with little success in the marketplace.²⁵⁹ However, cable and satellite television programming are examples of technically protected content that has met with commercial success.

Despite the mixed market results of technically protected content, interest in technical protection measures as a way of controlling access to and uses of digital forms of copyrighted works has grown considerably since the mid-1990's. The motion picture industry is the first copyright industry to successfully mass-market technically protected copies of digital content. Digital versatile disks (DVD) movies are protected by a technology known as the "Content Scrambling System" (CSS) that uses an authentication protocol to enforce country or region coding embedded in disks and players as well as an anti-copying mechanism.²⁶⁰ The motion picture industry has persuaded manufacturers of equipment to make players conforming to CSS so that the technical controls built into DVDs will be enforced.²⁶¹ The sound recording industry has been working on a Secure Digital Music Initiative (SDMI) to embed technical controls in digital sound recordings that would be "read" and enforced by players.²⁶² The publishing industry is hoping to develop secure e-books.²⁶³ Some technically protected content is already being delivered to consumers without the distribution of copies, such as by "streaming" of audio or video files over the Internet.²⁶⁴ More elaborate plans to build a "celestial jukebox" through

²⁵⁷ See, e.g., *Lochner Revisited*, supra note 248, at 520-25, giving Borland Int'l as an example of a new entrant willing to sell unprotected software to acquire market share from the market leader Lotus 1-2-3 that was selling copy-protected software, id. at 521, n. 221.

²⁵⁸ 17 U.S.C. sec. 1001 et seq.

²⁵⁹ See, e.g., *Lochner Revisited*, supra note 248, at 525-26.

²⁶⁰ See, e.g., *Universal City Studios, Inc. v. Reimerdes*, 111 F.Supp.2d 294, 308 (S.D.N.Y. 2000), aff'd, 2001 U.S. App. LEXIS 25330 (2d Cir. 2001).

²⁶¹ The DVD Copy Control Association (DVD-CCA) licenses CSS, certain patent rights necessary to make DVD players, and other knowhow to equipment manufacturers. See *DVD Copy Control Ass'n v. McLaughlin*, 2000 WL 48512 (Cal. Sup. Ct. 2000), rev'd sub nom. *DVD-CCA v. Bunner*, 201 D.A.R. 11709 (Cal. App. 6th Dist. 2001).

²⁶² See, e.g., Secure Digital Music Initiative website at <http://www.sdmi.org>.

²⁶³ See, e.g., Mark Stefik, *Shifting the Possible: How Trusted Systems and Digital Property Rights Challenge Us to Rethink Digital Publishing*, 12 Berkeley Tech. L.J. 137 (1997); Charles Clark, *The Answer to the Machine is In the Machine*, in *THE FUTURE OF COPYRIGHT IN A DIGITAL ENVIRONMENT* (P. Bernt Hugenholtz, ed. 1996) (discussing interest among publishers of scientific, technical, and medical publishers in electronic copyright management systems).

²⁶⁴ See, e.g., *RealNetworks, Inc. v. Streambox, Inc.*, 2000 U.S. Dist. LEXIS 1889 (W.D. Wash. 2000) (discussing streaming technology).

which consumers could order a wide range of technically protected digital content are also underway.²⁶⁵ One scholar believes that a fundamental transition for copyright industries is underway: from owning copies to experiencing works.²⁶⁶

Technical protection systems provide new opportunities for content owners to protect commercially distributed copyrighted works against unauthorized uses. They enable new business models, and importantly, they reduce the need to rely on the law of copyright to regulate uses of digital content in the hands of consumers.²⁶⁷ Technical protection systems are not, in themselves, fail-safe measures. What technology can do, another technology can undo. Some hackers regard technical measures as a challenge to be surmounted.²⁶⁸ Some computer scientists view them as suitable subjects for research.²⁶⁹ Those intent on infringing copyrights may also be motivated to break technical protections that rights holders use to protect their works.²⁷⁰ Reverse engineering is a necessary step in the undoing of any technical protection measure.

B. Circumstances Leading Up To the DMCA Rules

Prior to enactment of the DMCA, the legality of circumvention of technical measures used to protect copyrighted works or circumvention technologies had received little attention from the law. One exception was a provision in the AHRA forbidding the manufacture of technologies, the primary purpose or effect of which was to circumvent the SCMS chip in DAT machines.²⁷¹ Also outlawed was the sale of so-called “black boxes” for decoding encrypted satellite cable television programming.²⁷² Only one copyright case had considered the legality of making and selling a program that “undid” another vendor’s copy-protection system.²⁷³ Vault made a copy-protection program, Prolok, that it marketed to commercial software developers for use in protecting mass-market copies of their programs. Quaid reverse-engineered Prolok to figure out how it worked and developed a program called Ramkey that circumvented the Prolok system. Vault sued Quaid for contributory copyright infringement alleging that purchasers of Ramkey would use it to infringe copyrights of Vault’s customers’ programs and harm the

²⁶⁵ See, e.g., PAUL GOLDSTEIN, *COPYRIGHT’S HIGHWAY: FROM GUTENBERG TO THE CELESTIAL JUKEBOX* (1994).

²⁶⁶ Jane C. Ginsburg, *From Having Copies to Experiencing Works: The Development of an Access Right in U.S. Copyright Law*, in *U.S. INTELLECTUAL PROPERTY: LAW AND POLICY* (Hugh Hansen, ed. 2000).

²⁶⁷ See, e.g., *Digital Dilemma*, supra note 252, at 79-95.

²⁶⁸ See, e.g., *Universal City Studios, Inc. v. Reimerdes*, 111 F.Supp.2d 294, 320 (S.D.N.Y. 2000).

²⁶⁹ See, e.g., Amy Harmon, *Group Says It Beat Music Security but Can't Reveal How*, *New York Times*, Jan. 15, 2001, <http://partners.nytimes.com/2001/01/15/technology/15TUNE.html>.

²⁷⁰ See, e.g., Testimony of Gail Markels, General Counsel and Senior Vice President, Interactive Digital Services Software Ass’n, Hearing Before the Subcommittee on Courts & Intellectual Property of the Comm. On the Judiciary on H.R. 2281 and H.R. 2280, 105th Cong., 1st Sess. (1997) at 215-16 (discussing circumvention technologies used to enable “piracy” of copyrighted works).

²⁷¹ 17 U.S.C. sec. 1002(c).

²⁷² 47 U.S.C. sec. 605(e)(4).

²⁷³ *Vault Corp. v. Quaid Software Ltd.*, 847 F.2d 255 (5th Cir. 1988). The court did not regard copies made in the reverse engineering process to be infringing. *Id.* at 261, 270

market for Vault's software.²⁷⁴ The court ruled against Vault because Quaid's product had a substantial noninfringing use, namely, enabling users to make back-up copies of programs as copyright law authorized them to do.²⁷⁵

In 1995, as part of its National Information Infrastructure Initiative, the Clinton Administration proposed amending copyright law to outlaw circumvention technologies in its "White Paper" on Intellectual Property and the National Information Infrastructure.²⁷⁶ Without anti-circumvention legislation, the White Paper expressed concern that copyright owners would not provide content for the NII because their works would be too vulnerable to widespread infringements.²⁷⁷ To give new assurances to copyright owners, it proposed a ban on making or distributing technologies, the primary purpose or effect of which were to circumvent technical protections for copyrighted works.²⁷⁸ No longer would the existence of a substantial noninfringing use shield a technology from the control of copyright owners.

The Clinton Administration proposed a similar rule for a draft copyright treaty scheduled for consideration at a 1996 diplomatic conference convened at the World Intellectual Property Organization (WIPO).²⁷⁹ The draft treaty's anti-circumvention provision, modeled on the White Paper proposal, proved controversial once the conference began.²⁸⁰ Diplomats eventually agreed upon a compromise provision directing member states to provide "adequate protection" and "effective remedies" against circumvention of technical protections,²⁸¹ leaving the details of implementation to national discretion.

In 1997, the Clinton Administration announced its support for anti-circumvention rules that were more expansive than the original White Paper proposal.²⁸² Under this new legislation, it would be illegal to circumvent a technical measure used by copyright owners to protect access to their works. This provision was widely criticized as too broad. In response to some of these concerns, Congress crafted several specific

²⁷⁴ *Vault*, 847 F.2d at 258. One of the interesting questions in *Vault* was whether the copy-protection firm had standing to complain about infringement of software protected by Prolok in view of its not being the holder of copyrights in that software. The appellate court ruled that Vault did have standing because "RAMKEY destroys the commercial value of PROLOK disks." *Id.* at 263.

²⁷⁵ 17 U.S.C. sec. 117, discussed in *Vault*, 847 F.2d at 263-67.

²⁷⁶ Bruce Lehman, Report of the Working Group on Intellectual Property, Intellectual Property and the National Information Infrastructure 230-34 (Sept. 1995)(cited hereinafter as "White Paper").

²⁷⁷ *Id.* at 230.

²⁷⁸ *Id.*, app. 1 at 6.

²⁷⁹ See Pamela Samuelson, *The U.S. Digital Agenda at WIPO*, 37 Va. J. Int'l L. 369, 411-12 (1997) (discussing U.S. treaty proposal and the draft treaty provision).

²⁸⁰ *Id.* at 413-15 (discussing controversy over anti-circumvention rules at the WIPO conference).

²⁸¹ WIPO Copyright Treaty, adopted Dec. 20, 1996, CRNR/DC/94, art. 11. A similar treaty pertaining to sound recordings was also adopted at the same diplomatic conference, and it has a nearly identical anti-circumvention provision. See WIPO Performances and Phonograms Treaty, adopted Dec. 20, 1996, CRNR/DC/95.

²⁸² See Prepared Statement of Bruce A. Lehman, Hearing Before the Subcomm. On Courts and Intell. Prop. of the Comm. on the Judiciary, on H.R. 2281 (WIPO Copyright Treaties Implementation) and H.R. 2280 (Online Copyright Liability Limitation Act), 105th Cong., 1st Sess., Sept. 16-17, 1997 (cited hereinafter as "Judiciary Hearings").

exceptions to the anti-circumvention rules and authorized the Librarian of Congress to create other exemptions in periodic rulemakings.²⁸³ Much in contention was the impact of the anti-circumvention rules would have on fair uses of copyrighted works.²⁸⁴ Major copyright industry representatives opposed any exception for fair uses. One publishing industry witness stated: “Fair use doesn’t allow you to break into a locked library in order to make ‘fair use’ copies of books in it, or steal newspapers from a vending machine in order to copy articles and share them with a friend.”²⁸⁵ Circumvention and tools used for circumvention were analogized to burglary and burglars’ tools.²⁸⁶ Powerful rhetoric of this sort seems to have persuaded Congress that a general ban on circumvention and circumvention tools was necessary to protect copyrighted works in the digitally networked environment. Had Congress instead understood the DMCA rules as anti-reverse engineering rules, the legislative debate might have ended with a more balanced result.

The DMCA now permits circumvention for seven purposes: 1) for legitimate law enforcement and national security purposes,²⁸⁷ 2) for achieving program-to-program interoperability,²⁸⁸ 3) for engaging in “legitimate” encryption research (but this is subject to many conditions that substantially limit its application),²⁸⁹ 4) for testing the security of computer systems (also subject to many conditions that substantially limit its application),²⁹⁰ 5) for enabling non-profit libraries, archives, and educational institutions to make purchasing decisions,²⁹¹ 6) for allowing parents to control their children’s use of the Internet,²⁹² and 7) for protecting personal privacy.²⁹³ Since then, the Librarian of Congress has decided that circumventing access controls should be lawful in two other circumstances: 8) when an access control system is broken and the circumventor has a

²⁸³ The evolution of this legislation is recounted in detail in Pamela Samuelson, *Intellectual Property and the Digital Economy: Why the Anti-circumvention Rules Need to Be Revised*, 14 Berkeley Tech. L.J. 519 (1999). See also JESSICA LITMAN, DIGITAL COPYRIGHT 89-150 (2001).

²⁸⁴ See, e.g., Prepared Statement of Douglas Bennett, Judiciary Hearings, supra note 282, at 240-44.

²⁸⁵ Id. at 208 (prepared statement of Allan Adler of the Association of American Publishers).

²⁸⁶ See, e.g., House Manager’s Report, at 5 (characterizing circumvention tools as “the digital equivalent of burglars’ tools”).

²⁸⁷ 17 U.S.C. sec. 1201(e).

²⁸⁸ 17 U.S.C. sec. 1201(f). The reverse engineering exception adopts the core holding of *Sega v. Accolade* in legitimating reverse engineering when necessary to achieving interoperability. However, it narrows *Sega v. Accolade* by restricting what can be done with information obtained during the reverse engineering process, id. 1201(f)(3), by designating interoperability as the only legitimate purpose for which reverse engineering may be done, and by restricting the exception to achieving program-to-program interoperability even though circumvention may be needed to achieve hardware-to-program interoperability or program-to-data interoperability.

²⁸⁹ 17 U.S.C. sec. 1201(g). Limitations to this exception are discussed infra Section VI-C.

²⁹⁰ 17 U.S.C. sec. 1201(j).

²⁹¹ 17 U.S.C. sec. 1201(d). This exception is of very limited utility to nonprofit libraries, archives and educational institutions.

²⁹² 17 U.S.C. sec. 1201(h).

²⁹³ 17 U.S.C. sec. 1201(i). This provision only applies if the user did not receive advance notice that the technical protection system would be collecting personal data. For a discussion of the implications of digital rights management systems technologies on user privacy, see Julie E. Cohen, *A Right to Read Anonymously: A Closer Look at “Copyright Management” in Cyberspace*, 28 Conn. L. Rev. 981 (1996). For an example of a privacy-intrusive use of technical protection measures that is not covered by this exception, see Samuelson, supra note 283, at 552-54.

right to access the material, and 9) when necessary to assess the effectiveness of a software filtering program to determine what sites it blocks.²⁹⁴

Neither expressly authorizes the making of a tool to accomplish such privileged circumventions, and indeed it is unclear the Librarian of Congress has the authority to do so.²⁹⁵ Four of the seven statutory exceptions to the act-of-circumvention rule lack express authorization to make tools to accomplish circumventions.²⁹⁶ This raises a question whether there is an implied right to make a tool to engage in privileged circumventions or whether Congress created meaningless rights.²⁹⁷

The DMCA anti-circumvention rules respond to copyright industry fears of uncontrolled infringements as to digital versions of their content (movies, music, and the like). Digital content is very cheap and easy to copy and distribute via digital networked environments, and hence, it is vulnerable to market-destructive appropriations.²⁹⁸ As the well-known cryptographer Bruce Schneier has observed, “[d]igital files cannot be made uncopyable, any more than water can be made not wet.”²⁹⁹ Although digital content can be scrambled, every known scrambling system has been hacked. According to Schneier, “nothing works against a dedicated and skilled hacker... including unlock codes, encryption, serial numbers, hardware devices, on-line verification, copy protection, file encryption and watermarking.”³⁰⁰ Schneier says that almost any protection will work against the average user, but *no* protection system will work against the power user, hacker, or professional pirate.³⁰¹

The view articulated by Schneier may or may not be overstated, but we shall take it at face value as it provides the strongest argument for anti-circumvention rules. Even

²⁹⁴ See Copyright Office, Exemption to Prohibition of Circumvention of Copyright Protection Systems for Access Control Technologies, 65 F.R. 64555, 64574 (2000), codified in 37 C.F.R. sec. 201.40(b) (2000).

²⁹⁵ The Librarian’s rulemaking authority seems to be limited under 17 U.S.C. sec. 1201(a)(1)(C) to developing exceptions to the act of circumvention rule of 1201(a)(1)(A). Yochai Benkler argues that the DMCA anti-circumvention rules are unconstitutional, in part because the Librarian’s authority is too constricted. See Yochai Benkler, *Free As the Air to Common Use: First Amendment Constraints on Enclosure of the Public Domain*, 74 N.Y.U. L. Rev. 354 (1999). Many scholars question the constitutionality of the DMCA anti-circumvention rules. See, e.g., Brief Amicus Curiae of Intellectual Property Professors, submitted to the Second Circuit Court of Appeals in *Universal City Studios, Inc. v. Reimerdes* (Jan. 26, 2001), available at

http://www.eff.org/IP/DMCA/MPAA_DVD_cases/20010126_ny_lawprofs_amicus.html.

²⁹⁶ For a discussion of this problem, see Samuelson, *supra* note 283, at 537-46

²⁹⁷ See, e.g., *id.* at 547. See also Digital Dilemma, *supra* note 252, at 175 (noting ambiguity in the DMCA as to whether there is an implied right to make a tool to engage in privileged circumventions).

²⁹⁸ See *id.* at 28-45.

²⁹⁹ Bruce Schneier, *The Futility of Digital Copy Protection* at 2 (on file with the author). Schneier is the Chief Technology Officer of Counterpane Internet Security, Inc., designer of the popular Blowfish encryption system, and author of six books, including *SECRETS AND LIES: DIGITAL SECURITY IN A NETWORKED WORLD* (2000).

³⁰⁰ Bruce Schneier, *The Natural Laws of Digital Content*, slides of presentation at a conference on “Digital Libraries: Digital Asset Management” held at the Institute of Mathematics and its Applications in Minneapolis, MN, February 12, 2001 (on file with the authors).

³⁰¹ *Id.*

so, we will argue that the DMCA rules are far more restrictive than is necessary to achieve the objectives Congress had in mind when it adopted the DMCA rules.³⁰²

C. An Economic Analysis of the DMCA Rules

Broadly speaking, the anti-circumvention rules have consequences for protection of, access to, and uses of digital content, and competition in creating and marketing technical protection systems. Protection of digital copyrights was, of course, the principal motivation for the DMCA anti-circumvention rules. However we argue that the anti-circumvention rules go further than necessary to accomplish the goal of protecting digital content, causing collateral harm that could be avoided. In particular, the rules may unduly impinge on fair and other noninfringing uses of digital content, on competition within the content industry, on competition in the market for technical measures, and on encryption and computer security research.

From an economic standpoint, it would be better to maintain the DMCA's prohibition on public distribution of tools designed to circumvent technical protection measures (TPMs), but to exempt individual acts of circumvention and private tool-making incidental to such circumventions.³⁰³ This is consistent with the original White Paper proposal, which did not recommend legislation to outlaw acts of circumvention, but only to outlaw the manufacture and distribution of circumvention tools.³⁰⁴ While our proposed anti-tool rule is narrower than the White Paper's proposal,³⁰⁵ it nevertheless focuses on the same risk for copyright owners. As reflected in Table 4, the essence of our argument is that the narrower rule would achieve the intended benefits for copyright owners, while reducing harms to fair uses and improving incentives to develop, improve and use technical protection measures.

³⁰² Schneier believes that the DMCA rules will, in the end, prove futile because the Internet is an inherently global communications medium. Even if the U.S. and some allies adopt similar anti-circumvention rules, such rules "would never have the global coverage [they] need[] to be successful." Schneier, *supra* note 299, at 2. Schneier does not believe that the Internet spells the death of copyright, but only that "[w]e need business models that respect the natural laws of digital content instead of fighting them." *Id.*

³⁰³ Of course this does not mean that an individual act of circumvention should exempt the circumventor from liability if it results in copyright infringement. Circumvention of access controls may also sometimes violate the Computer Fraud & Abuse Act, 18 U.S.C. sec. 1030. We reserve judgment on whether an even better rule would be tort liability for distribution of circumvention tools rather than criminality liability.

³⁰⁴ See White Paper, *supra* note 276, appendix 1 at 6.

³⁰⁵ In particular, the White Paper would have outlawed technologies, the primary purpose or effect of which was circumvention. See *id.* Computer industry groups objected to the primary effect language in the White Paper's proposal because it put firms at risk if customers used products to circumvent, even if they were not designed to do so. A better rule is one that focuses on what the technology was designed to do, as our proposal would do. Our proposal would add a qualification about technologies that pose a high risk of facilitating infringement, as we believe that the DMCA anti-tool rules have sometimes been invoked where there is no danger of copyright infringement. See, e.g., *Sony Computer Entertainment of America, Inc. v. Gamemasters*, 87 F.Supp.2d 976 (N.D. Cal. 1999)(DMCA violation held to block sale of complementary product to technically protected game).

Table 4
Social Calculus of Reverse Engineering of Technically Protected Content:
Prior to the DMCA, Under the DMCA, and With Narrower Anti-Tool Rule

	Pre-DMCA	DMCA	Narrower Anti-Tool Rule (i.e., ban on tool distribution)
Incentives to develop content	low	high	moderate/high
Price of content	low	high	moderate
Fair uses of content	high	low	moderate
Expenditures on TPMs by content providers	high	moderate	high
Incentives to develop and improve TPMs	moderate	low	high
Wasted costs	high	low	low

1. Protecting Copyright

As is apparent from the legislative history, Congress's concern in enacting the DMCA was to protect copyrights in digital content. Without technical protections, digital content is vulnerable to uncontrolled copying. Technical protections generally do not prevent copying, but only make the digital content uninterpretable without authorized use of a key or detection of a watermark.³⁰⁶ An alternative to authorized use of a key is unauthorized decryption or circumvention, which involve reverse engineering. Decryption and circumvention are costly and difficult, and this is a significant check on the threat to copyright owners.

Most users have neither the inclination nor ability to circumvent a TPM.³⁰⁷ A potential infringer will only infringe rather than buy a legitimate copy if the cost of

³⁰⁶ For a discussion of technical protection measures that content owners are using or planning to use to protect their works, see, *e.g.*, *id.* at 152-73, Appendix E; Daniel J. Gervais, *Electronic Rights Management and Digital Identifier Systems*, J. Electronic Publishing, March 1999, available at <http://www.press.umich.edu/jep/04-03/gervais.html>. As Professor Lessig points out, the computer code that serves as a rights management technology is a kind of private governance system. See, *e.g.*, LAWRENCE LESSIG, *CODE AND OTHER LAWS OF CYBERSPACE* (2000).

³⁰⁷ See, *e.g.*, James R. Davis, *On Self-Enforcing Contracts, the Right to Hack, and Willfully Ignorant Agents*, 13 Berkeley Tech. L.J. 1145, 1147 (1998) (pointing out that most users won't have sufficient skill to circumvent to make fair uses). See also Richard J. Gilbert and Michael L. Katz, *When Good Value*

circumventing the TPM is less than the price of the copy. Content providers will take account of the potential for circumvention in setting their prices. As compared to the DMCA, content providers have an incentive to moderate their prices under the narrower rule, and also to employ effective TPMs.

The DMCA gives no incentive for the content providers to moderate their prices, and gives little incentive to employ effective TPMs. The DMCA allows for criminal penalties in cases of individual acts of willful circumvention and infringement.³⁰⁸ A circumventor would seem to be in jeopardy of criminal penalties even if the circumvention is trivial. Fear of such penalties, not the TPM, deters infringement. Under the DMCA, any trivial TPM may suffice, because circumventing a TPM raises the specter of criminal prosecution. Thus, the stringent penalties under the DMCA for individual acts of circumvention could have the odd consequence of *reducing* reliance on TPMs, as compared to the situation before the DMCA was enacted, and as compared to the narrower rule we propose. By reducing the market for effective TPMs, the DMCA also reduces the incentive to develop them and improve them, as we will discuss below.

Table 4 reflects these arguments. The price of copyrighted content susceptible to technical protection is likely to be highest under the DMCA, and lowest without any such legislation. The narrower anti-tool rule helps to enforce copyrights, but the price of content under this narrower rule is constrained by the threat of circumvention and infringement in a way that can be modified by the copyright holder in his choice of TPM.

Table 4 also shows that content providers' expenditures on TPMs will be higher under the narrower rule than under the DMCA, and probably highest with no legislation at all. Under a pre-DMCA regime of no prohibitions on circumvention, costs will likely be wasted on a measures-and-countermeasures war. Anti-circumvention rules may curb this war, but as explained above, the DMCA goes too far. It protects content owners without encouraging them to use really effective TPMs.³⁰⁹ A narrower anti-tool rule could both curb the measures-and-countermeasures war, and also encourage content providers to use effective TPMs for protection. There is, of course, a sense in which all expenditures on TPMs are "wasted", at least by comparison to an idealized world in which intellectual property is automatically respected. But in Table 4, we have separated "expenditures on TPMs" from "wasted costs." The latter reflects the cost of a measures-and-countermeasures war that can be avoided by appropriate circumvention rules.

Chains Go Bad: The Economics of Indirect Liability for Copyright Infringement, 52 *Hastings L.J.* 961, 982 (2001) (noting difficulties with regulating acts of circumvention).

³⁰⁸ 17 U.S.C. sec. 1204. In addition to being willful, an act of circumvention must also have a commercial purpose. We note that all infringements that displace a purchase will involve commercial harm to the copyright holder. But if "commercial purpose" is interpreted by courts to exclude infringement for personal use, then the criminalized enforcement of the DMCA would be less worrisome.

³⁰⁹ As Professor Peter Swire has observed, "After last week's events [the destruction of the World Trade Center towers by hijacked airplanes], it is less tolerable to have a legal regime that encourages weak computer security and makes it illegal to push companies toward stronger security." Email communication from Peter Swire, Sept. 14, 2001 (on file with the authors).

Content providers would likely spend more on TPMs under the narrower rule than under the DMCA, but we do not view this as a reason to prefer the DMCA. As we have explained, the DMCA protects rightholders by increasing the penalties for copyright infringement, not by encouraging the use of TPMs. We contend that if Congress wants to strengthen copyright enforcement by criminalizing infringement, then they should do it straightforwardly, rather than through the back door of the DMCA. While the narrower rule we propose is likely to increase the sums that content providers spend on TPM's, it avoids unnecessary criminalization of copyright infringement.

Our proposal for a narrower rule still maintains that the *public distribution* of circumvention tools should be prohibited. Otherwise, a single reverse engineer could induce widespread infringement by distributing the tool.³¹⁰ As Bruce Schneier puts it, “automation allows attacks to flow backwards from the more skilled to the less skilled.”³¹¹ In our view, that is the real threat that undermines the efficacy of TPM's, and should be kept in check. It is also worth noting that the rule against distribution of a circumvention tool is more likely enforceable, because detectable, than a rule against individual acts of circumvention.³¹²

2. Casualties of the DMCA: Fair Use and Competition

The main premise underlying the DMCA act-of-circumvention rule is that circumvention will overwhelmingly be undertaken for purposes of infringement. We dispute that premise. As the nine exceptions to this rule demonstrate, there are many reasons to circumvent technical protections that have nothing to do with copyright infringement. We note that three of the nine exceptions—those permitting reverse engineering to achieve interoperability among programs, encryption research, and computer security testing—are principally aimed at promoting follow-on innovation, either by permitting development of new products or improving products that already exist. The other six recognize that reverse engineering of technically protected digital content, such as reverse analysis of filtering software to discern what sites it blocks and decryption incidental to law enforcement and national security activities, may be reasonable and do not undermine copyright protection.

There are, however, many other reasons for reverse analysis of technical protections that promote follow-on innovation.³¹³ These include: 1) locating, assessing, and fixing bugs in software, 2) analyzing software to understand how to add additional

³¹⁰ The premise in this argument is that, although *creating* a circumvention tool is time-consuming and costly, the *use* of the tool is not. We notice, however, that notwithstanding the widespread availability of DeCSS, sales of DVD movies remain very strong and the motion picture plaintiffs in the *Reimerdes* case were unable to identify a single act of infringement of their movies attributable to the use of DeCSS. Thus, at least in the short run, and possibly in the long run, there may be impediments to *using* circumvention tools. Such impediments would also protect content owners, and thus render the restrictions on distribution unnecessary.

³¹¹ Schneier, *supra* note 257.

³¹² See Gilbert and Katz, *supra* note 307, at 982-83.

³¹³ Most of these examples and those in the following paragraphs occurred to us as we discussed the DMCA anti-circumvention rules; a few were suggested by others. Samuelson had previously identified some in a previous article on the DMCA. See Samuelson, *supra* note 283.

features, 3) understanding the internal design of a technical protection measure for research purposes, 4) understanding its internal design to develop a competing product, 5) understanding its internal design in order to make a compatible product, such as an alternative non-software platform, 6) analyzing a technical measure to enable interoperability with data, and 7) enabling critical commentary on a technically protected movie by taking fair use clips from it.³¹⁴

There are also many reasons to reverse engineer technical protection measures to enable other reasonable follow-on uses of technically protected digital content: 1) analysis of technical measures used to hide infringing copies of copyrighted works, 2) analysis of technical measures used to hide stolen trade secrets or other confidential information, 3) analyzing a virus program wrapped in a technical measure, 4) creating backup copies of software or data, 5) restoring a rightful copy after the crash of one's hard drive, 6) preserving information (e.g., evidence of some illegal activity), 7) preventing surveillance of a licensee's business activities, 8) preventing technical "self-help" measures from being wrongfully invoked, 9) bypassing country codes in a product so one can play a DVD movie for which one has already paid the standard fee on one's DVD player, 10) bypassing controls that prevent users from fast-forwarding through a movie, and 11) making other fair uses, such as by excerpting clips from technically protected movies to demonstrate that a particular word ("redskins") has been used in a derogatory fashion.³¹⁵

It is also worth pointing out that although circumvention of copy-control measures is not illegal under the DMCA, courts have, in essence, made it illegal by interpreting copy controls as "access controls," circumvention of which is banned under the DMCA. One court, for example, has declared that the Content Scrambling System

³¹⁴ Even Judge Kaplan has admitted that the fair uses excluded by technical protections are "remarkably varied." *Reimerdes*, 111 F. Supp.2d at 337-38 (giving examples). This judge concluded that the impact of the DMCA rules on fair use would be negative but "probably only to a trivial degree," *id.* at 337, because fair uses could be made of analog versions of movies, even if not of DVDs, and because some skilled technologists could make fair uses of DVD movies even if most people could not. An obvious flaw in the latter reason is that the skilled person would have to make a DeCSS equivalent in order to make fair uses of a DVD, which would seem to run afoul of section 1201(a)(2). It seems unreasonable to require a fair user to buy two copies of a movie instead of one to make fair uses or to relegate fair users to an inferior format. In addition, the former rationale ignores that analog VCR movies are also protected by technical measures and it would seemingly violate 1201(b) to make a tool to engage in fair uses of analog versions of movies. However, the Second Circuit Court of Appeals affirmed Judge Kaplan's rulings on Corley's fair use defenses in *Universal City Studios, Inc. v. Reimerdes*, 2001 U.S. App. LEXIS 25330 (2d Cir. 2001).

³¹⁵ Commentators differ in their views about the effects of the DMCA on fair uses. Some assert that Congress DMCA rules preclude fair uses. See, e.g., David Nimmer, *A Riff on Fair Use in the Digital Millennium Copyright Act*, 148 U. Penn. L. Rev. 673 (2000). Others find some basis in the DMCA for preserving fair uses as to technically protected works. See, e.g., Samuelson, *supra* note 283, at 540. See also Julie E. Cohen, *Copyright and the Jurisprudence of Self-Help*, 13 Berkeley Tech. L.J. 1089 (1998). Still others believe that the DMCA would be unconstitutional if it foreclosed fair uses. See, e.g., Benkler, *supra* note 295; Ginsburg, *supra* note 266; Neil Netanel, *Locating Copyright Within the First Amendment Skein*, 54 Stan. L. Rev. 1 (2001). See also Lessig, *supra* note 307, at 132-38 (characterizing as latent ambiguity whether the U.S. Constitution requires limitations on copyright, such as fair use)

(“CSS”) used to protect movies on DVDs is an access control.³¹⁶ As a consequence, purchasers of DVD movies can only play them on devices licensed by DVD-CCA, even if they would prefer to watch them on a Linux player. DVD movies can only be played on a device with the same country code as the movie. Another court characterized a country-coding scheme embedded in a mass-marketed videogame as an access control, thereby making it illegal to use lawfully purchased games on players with different country codes.³¹⁷ Without taking a position on the legitimacy or economic soundness of country-coding,³¹⁸ we wish to point out that these applications of the DMCA dramatically alter buyers’ rights, and we think further debate on these issues is warranted.

Some of the restrictions on use imposed by the DMCA, and overcome with the narrower rule, are not really “fair use” in the sense of copyright. However all these uses may lead to follow-on innovation. The DMCA inhibits fair uses and other uses by inhibiting access. Our proposed narrower anti-tool rule gives some opportunity to make fair uses that the copyright owner might want to prevent, and is therefore more likely to support follow-on innovations and reasonable uses, as reflected in Table 4.

A narrower anti-tool rule might also prevent anticompetitive uses of the DMCA by content providers. In the past three years, plaintiffs have asserted violations of the DMCA rules in order to exclude competitors from the marketplace,³¹⁹ to control the market for complementary products,³²⁰ and to facilitate their preferred market allocation

³¹⁶ *Reimerdes*, 111 F. Supp.2d at 317-18. This decision was recently affirmed in *Universal City Studios, Inc. v. Reimerdes*, 2001 U.S. App. LEXIS 25330 (2d Cir. 2001).

³¹⁷ *Sony Computer Entertainment America Inc. v. Gamemasters*, 87 F. Supp.2d 976 (N.D. Cal. 1999).

³¹⁸ There is a substantial international debate about whether the sale of intellectual property products in one nation should “exhaust” the rightsholders’ exclusive distribution rights throughout the world or whether rights should only be exhausted in the nation or region in which they were sold. Country codes embedded in software, games or DVDs are designed to enforce national or regional exhaustion preferences of the rightsholder. The economics of international vs. national or regional exhaustion are complex and as yet unresolved. For a discussion of the issues, see, e.g., Vincent Chiappetta, *The Desirability of Agreeing to Disagree: The WTO, TRIPS, International IPR Exhaustion, and A Few Other Things*, 21 Mich. J. Int’l L. 333 (2000).

³¹⁹ See, e.g., *Sony Computer Entertainment of America Inc. v. Gamemasters*, 87 F. Supp.2d 976, 982 (N.D. Cal. 1999)(successful 1201 claim against Game Enhancer software that competed with Sony’s Game Shark software). Sony also asserted anti-circumvention claims against Connectix, Inc. and Bleem, Inc. because both firms make emulator programs that did not read the anti-copying technology in Sony games. These emulator programs compete with PlayStation in the platform market. See Samuelson, supra note 269, at 556-57 (discussing Sony’s anti-circumvention claim against Connectix). See also Testimony of Jonathan Hangartner, attorney for Bleem, Inc., at Copyright Office Hearings on Anti-Circumvention Rules, at 224-32, held at Stanford University, May 19, 2000, available at <http://www.loc.gov/copyright/1201/hearings/1201-519.rtf> (discussing Sony’s anti-circumvention claim against Bleem and implications of 1201(a)(1)(A) going into effect for future Sony anti-circumvention claims against Bleem).

³²⁰ *Gamemasters*, 87 F. Supp.2d at 987-88 (Game Enhancer software that interoperated with Sony PlayStation games held to violate 1201 because it bypassed Sony country coding); *RealNetworks, Inc. v. Streambox, Inc.*, 2000 U.S. Dist. LEXIS 1889 (W.D. Wash. 2000) (Streambox VCR software, designed to interoperate with RealNetworks software, held to violate the anti-circumvention rules). See also *Reimerdes*, 111 F. Supp.2d at 320 (giving no weight to claim that DeCSS was intended to enable development of Linux platform for playing DVD movies).

and pricing strategies.³²¹ One commentator, disturbed by this trend, recommends development of a concept of misuse of DMCA rights akin to the misuse doctrines of patent and copyright law doctrines to thwart competitively harmful activities.³²²

Joint ownership of a proprietary technical protection system by major content providers could conceivably allow them to leverage their market power as to content into the market for equipment. For example, the motion picture industry controls the DVD player industry by its joint ownership of patent rights necessary to making DVD players; one of the conditions of this license is installation of CSS.³²³ More recently, the recording industry has sought to leverage its market power over digital music into the market for players, through the Secure Digital Music Initiative (SDMI). The goal of SDMI is to develop standard digital watermarks for digital music. The watermark must be detected by software in the player before the music can be heard.³²⁴ In both examples, players and content become a “system” much like the operating systems and applications software discussed in Section IV. In the player-movie/music systems, entry into the player market is foreclosed, in part because of the DMCA rules, which essentially make the interface proprietary.³²⁵ In the absence of legislation mandating installation of

³²¹ See, e.g., Michael Owen-Brown, Regulator challenges DVD zones, Australian Times (May 24, 2001), available at http://www.news.com.au/common/story_page/0,4057,2032464%25E421,00.html (Australian competition and consumer protection authorities investigating DVD country coding because of market allocation and discriminatory pricing impacts). See also <http://www.europa.eu.int/eur-lex/en/dat/2001/ce053/ce05320010220en01570157.pdf> and <http://www.europa.eu.int/eur-lex/en/dat/2001/ce053/ce05320010220en01580159.pdf> (EU competition concerns about country coding). See also <http://www.2600.com/news/display.shtml?id=541> (reporting on a raid of video stores in Gothenburg and Stockholm because they were selling imported DVD's with the “wrong” country code).

³²² Professor Dan Burk of the University of Minnesota Law School has a work in progress on misuse of DMCA anti-circumvention rights. Personal email communication from Dan Burk to Pamela Samuelson, dated Sept. 19, 2001.

³²³ See *Reimerdes*, 111 F.Supp.2d at 319. As a close reading of *Reimerdes* reveals, any firm that wants to make a DVD player needed to get a license from DVD-CCA to do so. *Id.* at 337. Although the court asserted that such licenses are “available to anyone on a royalty-free basis and at a modest costs,” *id.*, such licenses, in fact, are only available “subject to strict security requirements,” *id.* at 310. This precludes an open source Linux player. Any effort to develop an unlicensed platform would require reverse engineering of CSS (as well as make a tool to do so). The motion picture industry would almost certainly claim that this is illegal under the DMCA.

Jointly established royalties have the potential to facilitate price collusion, although there is no evidence that this has yet happened.

³²⁴ For a description of the SDMI watermarks and their intended uses, see SDMI challenge FAQ at <http://www.cs.princeton.edu/sip/sdmi/faq.html>.

³²⁵ In the software context, the market power constrained by reverse engineering constrained lies in the platform provider because of its control over APIs. In the digital entertainment context, the market power is chiefly wielded those who are rightsholders in the applications market. The economics of interoperability is the same, although the DMCA rules change the legal analysis significantly at least when firms want to develop alternative platforms to interoperate with digital data. The reverse engineering exception in 1201 only applies to program-to-program interoperability. In one decision, a judge interpreted this provision as inapplicable because technically protected DVD movies are not “programs,” but rather data. See *Universal City Studios, Inc. v. Reimerdes*, 82 F. Supp.2d 211,217-18 (S.D.N.Y. 2000). Because there are programs on DVDs as well as data, we question this ruling.

technical controls,³²⁶ the market power that is implicitly facilitated may be the only way to ensure that highly protected products will enjoy success in the marketplace.³²⁷

3. Competition in the Market for Technical Protection Measures

The incentive to develop and improve TPMs depends on market conditions, which in turn depends on whether copyright owners need TPMs to enforce copyrights. Thus the two main welfare considerations, namely, the effect of the anti-circumvention rules on TPMs, and the effect of the rules on copyright protection, are intertwined. We have argued that our proposed narrower anti-circumvention rule will increase the demand for effective TPMs, which will increase the incentive to develop and improve them.

The super-strong protection of the DMCA not only erodes incentives to *use* TPMs, but it also erects barriers to entering the market to supply them. The DMCA creates an extremely strong form of trade-secret-like protection for technical protection measures far beyond that provided by any other law. Ordinarily, an unpatented product, such as a technical measure, would be subject to reverse engineering and competition. As in the traditional manufacturing context, the vulnerability of unpatented products to reverse engineering limits market power in a competitively healthy way. The DMCA rules effectively insulate makers of technical protection measures from competitive reverse analysis, and that could be avoided by the narrower rule.

The narrower anti-tool rule would also enhance the ability of researchers to learn from each other. The DMCA may inhibit research and hence follow-on innovation in technical measures because it limits the ability of researchers to learn from their predecessors. A reverse engineer who discovers a problem with another firm's technical measure, and offers suggestions about how to improve it, is at risk of getting indicted on criminal DMCA charges, rather than being offered a commercial or academic opportunity to improve the product.

³²⁶ Senators Hollings and Stevens have announced their intent to introduce legislation to mandate installation of technically protections in future digital technologies. See, *e.g.*, Declan McCulloch, *New Copyright Bill Heading to DC*, Wired News, Sept. 7, 2001, available at <http://www.wired.com/news/politics/0,1283,46655,00.html> (discussing the Security Systems Standards and Certification Act that would mandate installation of standard technical protection measures in all interactive digital devices). Enactment of this legislation would foreclose the possibility of marketplace competition between protected and unprotected devices (except perhaps as regards used computers and other digital technologies).

³²⁷ Given a choice, consumers generally prefer unprotected products to protected products in part because technical protection measures often make products more difficult and inconvenient to us. See, *e.g.*, Digital Dilemma, *supra* note 255, at 87-88, 154. See also Anna Wilde Matthews, *AntiPiracy Tools in CDs Can Interfere With Playback*, Wall St. J., Nov. 29, 2001, at B1. As mentioned *supra* note 257 and accompanying text, marketplace competition among software developers led to the abandonment of copy-protection systems for software. Economists Carl Shapiro and Hal Varian assert that “[t]rusted systems, cryptographic envelopes, and other copy protection schemes have their place but are unlikely to pay a significant role in mass-market information goods because of standardization problems and competitive pressures.” CARL SHAPIRO & HAL VARIAN, *INFORMATION RULES* 102 (1998).

Reverse engineering lies at the very heart of encryption and computer security research: “[T]he science of cryptography depends on cryptographers’ ability to exchange ideas in code, to test and refine those ideas, and to challenge them with their own code. By communicating with other researchers and testing one another’s work, cryptographers can improve the technologies they work with, discard those that fail, and gain confidence in technologies that have withstood repeated testing.”³²⁸ A recent report of the National Academy of Sciences observes that “[r]egulating circumvention must be done very carefully, lest we hobble the very process that enables the development of effective protection technology.”³²⁹ The report identifies some key ambiguities in the DMCA’s anti-circumvention rules that put encryption and computer security researchers at risk.³³⁰ These assertions apply as much to commercial research as to academic research.

In the academic arena, the chilling effects of the DMCA on encryption and computer security research have already surfaced after the arrest of Russian programmer Dmitri Sklyarov who wrote a program capable of bypassing a technical protection measure in Adobe’s e-book software³³¹ and threats of litigation against Princeton computer scientist Edward Felten and his colleagues after he and his colleagues wrote a paper about flaws they discovered in digital watermarks that the recording industry planned to use to protect digital music.³³² Although the DMCA provides some room for

³²⁸ Brief of Amici Curiae of Dr. Steven Bellovin, Dr. Matt Blaze, Dr. Dan Boneh, Mr. Dave Del Torto, Dr. Ian Goldberg, Dr. Bruce Schneier, Mr. Frank Andrew Stevenson, and Dr. David Wagner, in *Universal City Studios, Inc. v. Reimerdes*, to the Second Circuit Court of Appeals, Jan. 26, 2001, available at <http://eol.law.harvard.edu/openlaw/DVD/NY/appeal/000126-cryptographers-amicus.html> (cited hereinafter as “Bellovin Amicus Brief”)

³²⁹ Digital Dilemma, *supra* note 255, at 173.

³³⁰ See, *e.g.*, *id.*, App. G (discussing ambiguities and other problems with the DMCA anti-circumvention provisions).

³³¹ See, *e.g.*, Robert Lemos, FBI nabs Russian expert at Def Con, ZDNet News, July 17, 2001, <http://www.zdnet.com/zdnn/stories/news/0.4586.5094266.00.html>. Instead of fixing the flaw in its software, Adobe asked the Justice Dept. to prosecute Dmitri Sklyarov, a Russian citizen who wrote the software in Russia (where development of such software is apparently legal) while he was in the U.S. at a conference.

³³² In September 2000 the Secure Digital Music Initiative (SDMI) issued a public challenge inviting skilled technologists to defeat digital watermarking technologies that SDMI had selected as candidate standards for protecting digital music. See “An Open Letter to the Digital Community” available at http://www.sdmi.org/pr/OL_Sept_6_2000.htm. SDMI offered to pay successful hackers \$10,000 per broken watermark. Princeton computer scientist Edward Felten and his colleagues decided to accept this challenge, although not to seek the prize money because SDMI was only willing to award the money to those who agreed not to reveal how they defeated the watermarks to anyone but SDMI. Felten and his colleagues instead wrote a paper for a scientific workshop on the results of their research about the SDMI watermarks. The paper was entitled “Reading Between the Lines: Lessons From the SDMI Challenge” and was scheduled for presentation at the Fourth International Information Hiding Workshop in Pittsburgh, Pennsylvania, on April 26, 2001. For further details, see SDMI challenge FAQ at <http://www.cs.princeton.edu/sip/sdmi/faq.html>.

An executive from Verance, the developer of one of the candidate technologies, and the Recording Industry Association of America (RIAA) found out about the paper and asked Felten to omit certain details about the weaknesses of the SDMI technologies. Felten and his coauthors decided that these details were necessary to support their scientific conclusions. SDMI and RIAA asserted that presentation of the paper at the conference or its subsequent publication in the conference proceedings would subject Felten, his coauthors, members of the program committee, and their institutions to liability under the DMCA, and made clear their intent to take action against the researchers unless they withdrew the paper. RIAA’s

encryption and computer security research, the exceptions for these activities are so narrowly drawn that neither seems to apply to Sklyarov or Felten.³³³ Consider, for example, that the encryption research exception does not apply to Felten because his research focused on digital watermarks that do not use encryption.³³⁴

This and other restrictions have caused prominent cryptographers to characterize the encryption research and computer security exceptions as “so parsimonious as to be of little practical value” as well as being based on a “fundamentally mistaken conception of cryptographic science.”³³⁵ The encryption research exception only applies, for example, if the researcher is employed or has been trained as a cryptographer,³³⁶ even though some brilliant breakthroughs have come from persons without such training.³³⁷ The exception is also only available if the researchers have sought permission from affected rightsholders before trying to reverse engineer an encryption technology.³³⁸ Researchers must, moreover, prove the necessity of their acts.³³⁹ And the exception may be

theory is that the presentation of the paper constitutes distribution of a circumvention tool in violation of 1201(b)(1). A copy of the RIAA letter to Professor Felten asserting that presentation or publication of the researchers’ paper would violate the DMCA is available at <http://cryptome.org/sdmi-attack.htm>.

Although convinced that they would be vindicated if the matter went to court, Felten and his coauthors reluctantly withdrew the paper from the April conference out of concern about the high costs of litigation. See announcement at <http://cryptome.org/sdmi-attack.htm>. Felten’s decision was widely reported in the press. See, e.g., David P. Hamilton, Professor Savors Being in the Thick of Internet Rows, *Wall St. J.*, p. B1, June 14, 2001; Charles C. Mann, *Secure-Music Group Threatens Researchers Who Plan to Publish on Hacking Success*, *Inside Mag.*, 4/22/01, available at <http://www.inside.com>. The Electronic Frontier Foundation agreed to represent Felten and his coauthors in an affirmative challenge to the RIAA and SDMI claim which seeks a judicial declaration that the paper does not violate the DMCA so that Felten can present the paper at a conference in August 2001. See *Felten v. RIAA Complaint*, available at http://www.eff.org/Legal/Cases/Felten_v_RIAA/20010606_eff_complaint.html. This complaint was recently dismissed because SDMI, RIAA, and Verance withdrew their objections to publication of this paper. See, e.g., John Schwartz, 2 Copyright Cases Decided in Favor of Entertainment Industry, *N.Y. Times*, Nov. 29, 2001, at C-4.

³³³ 17 U.S.C. sec. 1201(g), 1201(j). If the SDMI watermarks are not access controls, the computer security testing exception would be inapplicable because it only permits making a tool to bypass an access control under 1201(a)(2), not making a tool to bypass other controls under 1201(b). Neither privilege applies to claims under 17 U.S.C. sec. 1202, a provision that protects copyright management information from alteration or removal. Verance had claimed that Felten violated section 1202 as well as 1201. Sklyarov does not qualify for either exception, even though he is a trained cryptographer, because the firm for which he works has sold copies of the bypassing software over the Internet, thereby distributing the tool beyond the scope of the exception. Also Sklyarov did not get Adobe’s permission before testing its e-book security, as section 1201(j) requires.

³³⁴ 17 U.S.C. sec. 1201(g).

³³⁵ Bellovin Amicus Brief, supra note 328. Problems with the overly narrow and ambiguous encryption and computer security exceptions to the DMCA are discussed in *Digital Dilemma*, supra note 255, at 174-75, Appendix G (2000).

³³⁶ 17 U.S.C. sec. 1201(g)(3)(B).

³³⁷ See SIMON SINGH, *THE CODE BOOK: THE SCIENCE OF SECRECY FROM ANCIENT EGYPT TO QUANTUM CRYPTOGRAPHY* (1999) (discussing the history of cryptography, and in particular, the contributions of Whitfield Diffie).

³³⁸ 17 U.S.C. sec. 1201(g)(2). The computer security exception requires that the researcher actually get, and not just ask for, permission to defeat the technical protection measure. 17 U.S.C. sec. 1201(j)(1).

³³⁹ 17 U.S.C. sec. 1201(g)(1), (g)(2)(B).

unavailable if the researcher publishes his or her results on the Internet, where they may be accessible to potential pirates.³⁴⁰

Encryption and computer security may be crippled if researchers are at risk of liability under the DMCA in the ordinary course of their research.³⁴¹ As we argued for SCPA, reverse engineering can facilitate competition for improvements. The right balance between facilitating improvements and protecting earlier innovators can be achieved by granting a kind of “leading breadth” to each innovation,³⁴² but not by prohibiting researchers from access to existant knowledge, as the DMCA does.

VI. Reverse Engineering as a Policy Lever

All intellectual property rights regimes—utility patent, plant variety protection, copyright, and SCPA—have certain policy levers in common, wielded to a greater or lesser extent. All establish, for example, a length of protection, a breadth of protection (sometimes legislated and sometimes evolving through caselaw interpretations), and some fair use or policy-based limitations on the scope of protection. By wielding the available policy levers appropriately, legal regimes can be made sensitive to the technological and industrial contexts they regulate so as to avoid either over-rewarding or under-rewarding innovators.

We conceive of the legal status of reverse engineering as one such policy lever. This policy lever is set differently in different legal contexts. Trade secrecy law, for example, exposes innovators to reverse engineering whereas patent law limits it to some degree.³⁴³ A rationale for this difference lies in the disclosure obligations that patent law imposes on innovators that trade secret owners avoid. For the traditional subject matters of copyright law, namely, artistic and literary works, reverse engineering has not been an issue because viewers and readers did not need to reverse engineer these works to understand them. Yet as copyright’s subject matter expanded to include computer software, reverse engineering became a significant policy issue in copyright law as well.³⁴⁴

The optimal setting for any given policy lever depends in part on how the other levers are deployed.³⁴⁵ Consider, for example, the interaction of reverse engineering

³⁴⁰ 17 U.S.C. sec. 1201(g)(3)(A). The encryption researcher must also provide affected copyright owners with the results of his or her research in a timely manner. 17 U.S.C. sec. 1201(g)(3)(D).

³⁴¹ See, e.g., Andrew W. Appel and Edward W. Felten, *Technological Access Control Interferes with Noninfringing Scholarship*, 43 Comm. ACM 21 (Sept. 2000); Bellovin Amicus, supra note 327; Pamela Samuelson, *Anti-Circumvention Rules Threaten Science*, 293 Science 2028 (Sept. 2001).

³⁴² See, e.g., O’Donoghue et al, supra note 140.

³⁴³ See supra notes 30-37 and accompanying text.

³⁴⁴ See supra Section IV-A.

³⁴⁵ There is an extensive economics literature on the interdependency of intellectual property policy levers. Most saliently, economics scholars have addressed the interaction of length and breadth. See, e.g., Richard Gilbert and Carl Shapiro, *Optimal Patent Length and Breadth*, 21 RAND J. Econ. 106 (1990); Paul Klemperer, *How Broad Should the Scope of Patent Protection Be?*, 21 RAND J. Econ. 113 (1990); Nancy T. Gallini, *Patent Length and Breadth with Costly Imitation*, 24 RAND J. Econ. 52 (1992). See also

rules and the length of protection. Outlawing decompilation of computer programs is inadvisable in part because of the long duration of protection that copyright provides to programs.³⁴⁶ If decompilation and disassembly were illegal, programs would be immune from an important source of competition for almost a century, which would likely impede innovation in the software industry. Such a rule would provide far more protection than necessary to protect innovative software firms against market-destructive appropriations.

Our study of reverse engineering in various industrial contexts leads us to two general conclusions. The first is that reverse engineering has generally been a competitively healthy way for second comers to get access to and discern the know-how embedded in an innovator's product. If reverse engineering is costly and takes time, as is usually the case, innovators will generally be protected long enough to recoup R&D expenses. More affirmatively, the threat of reverse engineering promotes competition in developing new products and constrains market power as well as inducing licensing that enables innovators to recoup R&D costs.

Second, we have found it useful to distinguish between the act of reverse engineering, which is generally performed to obtain know-how about another's product, and what a reverse engineer does with the know-how thereby obtained (e.g., designing a competing or complementary product).³⁴⁷ The act of reverse engineering rarely, if ever, has market-destructive effects and has the benefit of transferring knowledge. Harmful effects are far more likely to result from post-reverse engineering activities (e.g., making a competing product with know-how from an innovator's product). Because of this, it may be more sensible to regulate post-reverse engineering activities than to regulate reverse engineering as such. This view is reinforced by difficulties of enforcement. Acts of reverse engineering typically take place in private and are more difficult to detect than post-reverse engineering activities (such as introducing competing or complementary products to the market). They are, as a consequence, less susceptible to effective regulation. In the discussion below, we distinguish between regulatory strategies aimed at acts of reverse engineering and those aimed at post-reverse engineering activities.

The bluntest way to deploy the reverse-engineering lever is to switch it "on" (making it legal) or "off" (making it illegal). Our study has revealed five more nuanced ways to deploy this lever: 1) regulating a particular means of reverse engineering,³⁴⁸ 2) establishing a breadth requirement for subsequent products,³⁴⁹ 3) using purpose- and necessity-based requirements for judging the legitimacy of reverse engineering,³⁵⁰ 4)

Maurer & Scotchmer, *supra* note 57, for the static context, and O'Donoghue, et al., *supra* note 140, for the cumulative context.

³⁴⁶ We agree with Graham and Zerbe on this point. See Graham & Zerbe, *supra* note 162, at 128-31.

³⁴⁷ The SCPA rules, for example, explicitly permit reverse engineering, but impose a burden on reverse engineers to invest in post-reverse engineering design work. See *supra* Section III-C. The DMCA rules, in contrast, include restrictions on acts of reverse engineering of technical protection measures. See *supra* Section V. The anti-tool rules of the DMCA go further in regulating preparatory activities for reverse engineering, namely, the making of tools for use in reverse engineering.

³⁴⁸ See Section II-C *supra*.

³⁴⁹ See Section III-C *supra*.

³⁵⁰ See Section IV-A *supra*.

regulating reverse engineering tools,³⁵¹ and 5) restricting publication of information discovered by a reverse engineer.³⁵²

We review these options in subsection A, for two reasons. First, they have been adopted in some industrial contexts and should be assessed for their economic reasonableness. Second, proposals for additional restrictions on reverse engineering may be made in the future. Legal decisionmakers may be better equipped to respond to such proposals if they understand how reverse engineering has been regulated in the past and under what conditions restrictions on reverse engineering are justifiable.

In subsection B, we observe that the existence of a legal right to reverse engineer may be so threatening to some innovators that they will endeavor to render the legal right moot through one of two strategies: 1) by requiring customers to agree not to reverse engineer their products, or 2) by configuring their products to make reverse engineering extremely difficult or impossible. Legal decisionmakers have the option of responding to such efforts by deciding not to enforce such contractual restrictions or by forcing disclosure of product know-how.

A. Ways to Regulate Reverse Engineering

1. Regulating a Market-Destructive Means of Reverse Engineering

When a particular means of reverse engineering makes competitive copying too cheap, easy or rapid, innovators may be unable to recoup R&D expenses. If so, it may be reasonable to regulate that means. Anti-plug mold laws, discussed in section II-C, are an example. Using a competitor's product as a "plug" with which to make a mold from which to make competing products permits competitive copying that is so cheap and fast as to undermine the incentives to invest in designing an innovative product. Restrictions on plug molding may restore adequate incentives to make such investments. Notwithstanding the Supreme Court's characterization of plug-molding as an efficient means of reverse engineering,³⁵³ we suggest that plug molding is better understood as an

³⁵¹ See Section V-B *supra*.

³⁵² See *supra* note 177. Our study also uncovered four other proposals to regulate reverse engineering in the software industry. One proposal was to allow decompilation or disassembly if done through a "clean room" process, that is, by separating the team assigned to reverse engineer another firm's program from the team that uses information provided by the first team in developing a new program. See, e.g., Laurie & Everett, *supra* note 162. Second, one decision would have allowed reverse engineering of a program for purposes of achieving present compatibility with the other firm's software, but not for purposes of achieving future compatibility. See *Atari Games Corp. v. Nintendo of Am., Inc.*, 975 F.2d 832 (Fed. Cir. 1992). Third, the legality of the second comer's reverse engineering efforts has sometimes undermined by a "taint" in the last stages of the reverse engineering process, as when defendant's lawyers lied to the U.S. Copyright Office in order to get a copy of plaintiff's source code that the innovator had filed when registering its claim of copyright. In essence, the Atari Games' engineers' efforts to reverse engineer from the code did not yield enough information, so the lawyers were sent to get source code listings on file in the Copyright Office so that they could get the additional information they needed to make compatible games. This inequitable conduct affected the court's view on Atari's fair use defense. *Id.* Fourth, the European Software Directive seems to give weight to establishing a "paper trail" to show the legitimacy of reverse engineering of software may be important. See Czarnota & Hart, *supra* note 177, at 84.

³⁵³ *Bonito Boats*, 499 U.S. at 160.

efficient means of reimplementing the original innovation. Plug molding has the potential to undermine an innovator's incentives without any offsetting social benefit of follow-on innovation because a plug-molder does not aim to learn anything that might lead to further innovation. Thus, one of the key benefits of reverse engineering will be lost if plug-molding is utilized to make competing products.

Another controversial act of reverse engineering was decompilation and disassembly of computer programs, discussed in Section IV.³⁵⁴ Some industry participants feared that reverse engineering would allow second comers to appropriate valuable internal design elements of programs. Decompilation and disassembly were eventually accepted as legal, in part because they require so much time, money and energy that the original developer is not significantly threatened. If reverse engineering actually occurs in face of these costs, it may enable the development of competitive interoperable products and erode the market power of industry leaders in a competitively healthy way.

Our advice to policymakers is this: Before banning a means of reverse engineering, require convincing evidence that this means has market-destructive consequences. Realize that existing market participants may be seeking a ban mainly because they wish to protect themselves against competitive entry. Any restriction on reverse engineering should be tailored so that it does not reach more than parasitical activities. For example, it may be sensible not to make the restriction retroactive, to require that innovations embody some minimal creativity, and/or to limit the duration of the ban.³⁵⁵ Another possibility is to outlaw market-destructive reimplementations of innovations, rather than banning reverse engineering as such. Alternatively, reverse engineers could be required to compensate rightsholders for research uses of the innovation aimed at development of follow-on innovation.³⁵⁶

2. A Breadth Requirement For Products of Reverse Engineering

Another policy option is to establish a breadth requirement for products developed after reverse engineering.³⁵⁷ If second comers must invest in some forward-engineering and not simply free-ride on the previous innovation by copying it exactly, the second comer's efforts are more likely to advance the state of technology, as well as to extend the second comer's development cycle so that the earlier innovator is still

³⁵⁴ The DMCA rule outlawing circumvention (that is, reverse engineering) of technical measures that control access to copyrighted works does not qualify under this category because it does not focus on a particular means. Yet, it too is a direct regulation of acts of reverse engineering. Although this rule has some exceptions (e.g., to enable program-to-program interoperability), Section V has explained why that ban is nevertheless too restrictive.

³⁵⁵ See supra notes 80-84 and accompanying text.

³⁵⁶ See, e.g., Eisenberg, supra note 40 (proposing compensation for research uses of patented research tools); Mueller, supra note 40 (accord); Manifesto, supra note 15 (proposing liability rule for reuses of industrial compilations of applied know-how in software); Reichman, *Green Tulips*, supra note 56 (proposing liability rules for subpatentable innovation).

³⁵⁷ See the discussion in note 141 above of how economists have treated breadth.

protected. The Semiconductor Chip Protection Act was our principal example.³⁵⁸ The SCPA permits intermediate copying of chip circuitry for purposes of study and analysis; it also permits reuse of some know-how discerned in the reverse engineering process. This is a useful boost to competitors designing integrated circuits. However, SCPA requires reverse engineers to design an “original” chip rather than simply making a clone or near-clone of the integrated circuit that was reverse engineered.³⁵⁹

Since SCPA allows later innovators to learn from earlier ones, while still allowing chip designers to recoup expenses, we think it is competitively healthy. More generally, we find merit in the idea of establishing a breadth requirement to ensure that reverse engineering leads to further advance, while still preserving enough market power so that innovator recoups costs, in markets where cloning the innovator’s product will be market-destructive.³⁶⁰ Again, policy makers should be wary of undocumented claims that reverse engineering is *per se* destructive.³⁶¹ Establishing breadth requirements may be unnecessary to protect the lead-time of innovators in many industries because the costliness and difficulties of reverse engineering and reimplementing may provide adequate protection.³⁶² The SCPA rules responded to specific perturbations in the semiconductor chip market that undermined lead-time.

While most legal regimes do not link the legitimacy of reverse engineering with technical advance, the software copyright case law may do so implicitly. In *Sega v. Accolade*, for example, the court’s perception that Accolade’s reverse engineering was legitimate rested in no small part on the defendant’s having developed a new, noninfringing program that promoted the very kind of progress that copyright law was intended to bring about.³⁶³ Nevertheless, a linkage between the legitimacy of reverse engineering and a breadth requirement in the software industry may be unnecessary for two reasons: First, decompilation and disassembly of programs are so difficult and time-consuming that second comers generally do not find it profitable to develop market-

³⁵⁸ A similar rule exists in the Plant Variety Protection Act, 7 U.S.C. sec. 2541, 2544. Use of a protected variety to develop a new variety is non-infringing as long as the subsequent variety itself qualifies as a distinct variety that qualifies for PVPA protection. In 1994 Congress limited application of this rule so that if the subsequent variety retains virtually the whole genetic structure of the earlier variety, the subsequent variety may infringe. See, e.g., Peter J. Gross, *Guiding the Hand that Feeds: Toward Socially Optimal Appropriability in Agricultural Biotechnology Innovation*, 84 Calif. L. Rev. 1395 (1996).

³⁵⁹ SCPA’s reverse engineering privilege may be instructive even if SCPA itself is flawed or no longer necessary for reasons discussed *supra* Section III-C.

³⁶⁰ There are, of course, important issues about how much progress should be required for the new product to be permissible, but the basic principle is sound: By prohibiting clones, but permitting reverse engineering to make improved products, each innovator is protected for some period against horizontal competition, but must eventually give way to a better product.

³⁶¹ See *supra* Section II-C.

³⁶² In addition, lead time can be governed by breadth. The length of each innovator’s dominance in the market is determined in part by how long it takes a rival to find a noninfringing improvement. O’Donoghue et al, *supra* note 140, refer to this lead time as the “effective patent life.” It may be shorter than the statutory patent life, and in this way, the possibly excessive reward granted by a 20-year term of patent can be modified endogenously by breadth.

³⁶³ See *supra* Section IV-A.

destructive clones in this way.³⁶⁴ Second, reverse engineering of software does not generally lead to the development of a competing product, but rather to the development of interoperable programs or to the fixing of software “bugs.” Breadth requirements seem most appropriate when the goal is development of a competing product.

3. Purpose- And Necessity-Based Criteria for Determining the Legitimacy of Reverse Engineering

A third way to deploy the reverse engineering policy lever is to judge its legitimacy based on its purpose or necessity.³⁶⁵ As with regulation of particular means, this approach focuses on the act of reverse engineering itself. Purpose-based rules assume that reverse engineering is sometimes socially beneficial and sometimes harmful, and at a deeper level, that society will benefit from a reverse engineer’s acquisition of some types of know-how embedded in commercially distributed products but not others. Necessity-based rules assume that societal resources should not be expended on reverse engineering if the information being sought is already available. It is worth noting that the legitimacy of reverse engineering has traditionally not depended on its purpose or necessity. For traditional manufactured items, the right to reverse engineer has been almost absolute.

Two examples of purpose- and necessity-based privileges from this study are *Sega v. Accolade* and its progeny that permit reverse engineering of computer software for the purpose of achieving interoperability,³⁶⁶ and the DMCA anti-circumvention rules that permit reverse engineering of access controls for some purposes, among them, achieving interoperability.³⁶⁷

We have mixed reactions to purpose- and necessity-based criteria for regulating reverse engineering. Of course it is true that the economic effects of reverse engineering depend on the reverse engineer’s purpose, and purpose-based reasoning is common in intellectual property law. A second comer’s purpose often determines whether he or she qualifies for an exception to or limitation on intellectual property rights.³⁶⁸ Copyright’s fair use doctrine, for example, gives considerable weight to the purpose of a fair-use claimant’s activities.³⁶⁹

One positive consequence of purpose-based rules is to induce knowledge-sharing through licensing or voluntary disclosure. European policymakers decided to legalize

³⁶⁴ To the extent decompilation results in an infringing program, copyright law already provides an adequate remedy. See, e.g., *E.F. Johnson Co. v. Uniden Corp. of Am.*, 623 F. Supp. 1485 (D. Minn. 1985) (infringement based not on decompilation but on copying of expressive aspects of program internals).

³⁶⁵ Judging the legitimacy of reverse engineering based on the person’s purpose for engaging in the activity or on the necessity of the reverse engineering would seem, in theory, distinct mechanisms for regulating reverse engineering. Because these two criteria have been linked in the regulation of reverse engineering in the software industry and in the DMCA, we will treat them together in this subsection.

³⁶⁶ *Sega*, 977 F.2d at 1522.

³⁶⁷ 17 U.S.C. sec. 1201 (d)-(i).

³⁶⁸ See *supra* notes 37, 40, and 164 accompanying texts (discussing the experimental use defense in patent law, the research exception in PVPA, and the fair use defense in copyright law).

³⁶⁹ 17 U.S.C. sec. 107 (preamble, subsec. 1).

decompilation of computer programs for purposes of achieving interoperability in order to make the threat of reverse engineering credible enough so that software developers would disclose interface information voluntarily or license it on reasonable terms.³⁷⁰ This would ensure that European software developers could enter software markets with interoperable products. European policymakers did not wish to encourage licensing of other program know-how, but rather sought to encourage second comers to do their own independent design work, and hence, they restricted the privilege to decompilation for interoperability.³⁷¹

A downside of purpose-based regulations is that if reverse engineering is not averted by licensing, wasteful litigation may be the only way to determine the reverse engineer's purposes. Antitrust law faces similar difficulties, as when a court must decide whether a certain defendant (say, Microsoft) engaged in certain acts for good purposes (e.g., integrating its browser into the operating system to benefit consumers) or bad purposes (e.g., trying to put Netscape out of business).³⁷² Moreover, reverse engineers may have multiple purposes, only some of which may be privileged by purpose-based rules.

We believe that purpose-based rules are better than necessity-based rules as a strategy for limiting reverse engineering. Necessity-based rules may be a trap for the unwary. For example, if a software developer offers to license interface information on terms a second comer deems unreasonable, reverse engineering may seem necessary to the second comer, but not to the prospective licensor. Similarly, a software developer may be willing to license a minimal amount of interface information, but not enough to make the program fully interoperable. Once again, whether reverse engineering is necessary is disputable. A further problem arises if the information is available in an obscure place unknown to the reverse engineer who, in ignorance, exposes himself to liability by going ahead with reverse engineering he believed to be necessary. Necessity-based rules would also seem to be largely unnecessary given that rational second comers would almost always prefer to avoid the expenses of reverse engineering if the desired information is available without it.

Finally, we observe that enumerating exceptions to a general prohibition has different consequences than enumerating exceptions to a general privilege. The DMCA has a general prohibition of reverse engineering of access controls that is subject to various purpose-based exceptions.³⁷³ This approach implies that reverse engineering of

³⁷⁰ See Official Commentary to the European Software Directive, reproduced in Czarnota & Hart, *supra* note 177, at 76-80.

³⁷¹ *Id.* at 79-80. The *Sega v. Accolade* decision, by contrast, regards decompilation as legitimate if done to get access to information that is unprotected by copyright law (e.g., algorithms or mathematical constants). See, e.g., *Sega*, 977 F.2d at 1526 (emphasizing the need to decompile to access unprotected aspects of programs). Purpose-based limits on reverse engineering may also protect developers against difficult-to-detect infringements or avoid wasteful expenditures on reverse engineering undertaken for harmful purposes (e.g., to develop virus programs).

³⁷² See, e.g., *U.S. v. Microsoft Corp.*, 243 F.3d 34, 84-96 (D.C. Cir. 2001)(review of conflicting views on Microsoft's purposes in integrating Internet Explorer into the Windows operating system).

³⁷³ See 17 U.S.C. sec. 1201(a)(1)(A) (general prohibition), (d)-(k) (purpose-based exceptions).

access controls for every other purpose is illegal. Given that the principal objective of the DMCA rule is to prevent copyright infringements, a more straightforward approach would have been to establish a general privilege in favor of reverse engineering, but to disallow it for purposes of enabling infringement copyrighted work. Because the DMCA adopts the more restrictive approach, a host of reasonable circumventions must be presumed illegal.³⁷⁴ Those who reverse engineer for unenumerated, but benign, purposes can only hope that their activities will escape the notice of the copyright industries and federal prosecutors.

4. Regulating Reverse Engineering Tools

The DMCA anti-circumvention rules are unique among the legal regimes we studied in regulating the development and distribution of tools for reverse engineering. This strategy does not regulate the act of reverse engineering or post-reverse engineering activities so much as preparatory activities necessary to engage in reverse engineering. For reasons given in section V, we think the DMCA's anti-tools rules are overbroad, but we recognize that these rules cannot be judged by the same considerations as we used in other industrial contexts. Our general assumption about reverse engineering in other contexts has been that once the proper boundaries of intellectual property are established, the property right will be enforced. The anti-tool rules, in contrast, are directed at the problem of enforcement.

The enforcement problem arises because digital content is very cheap and easy to copy. To overcome this, the entertainment industry is increasingly using technical measures to protect their content from unauthorized access and use. Circumvention undermines this strategy. Since circumvention tools are essential to reverse engineering of these technical measures, the entertainment industry persuaded Congress to outlaw circumvention tools. We agree that there are some good economic arguments for regulating trafficking in anti-circumvention technologies. Without ready access to circumvention tools, both large- and small-scale infringements may be prevented. It is, moreover, easier to detect and police a public market in circumvention technologies than to control private acts of circumvention and copying.³⁷⁵ Nevertheless, we have argued that the anti-tool rules of the DMCA are defective because they reach many activities that have little marginal value for enforcement purposes. Overbroad anti-tool rules are also harmful because they have provided copyright owners with a potent weapon for excluding competitive or complementary products from the market.³⁷⁶ They also facilitate the ability of copyright owners to leverage their market power in content into the equipment market.

5. Restricting Publication of Information Discovered By a Reverse Engineer

A fifth policy option is to allow reverse engineering but forbid publication or other disclosure of information obtained thereby. For the most part, the law has not had

³⁷⁴ See supra Sections V-C.

³⁷⁵ See, e.g., Gilbert & Katz, supra note 307, at 982-83.

³⁷⁶ See supra Section V-C.

to address this issue because reverse engineers have generally had little incentive to publish or otherwise disclose information they learn from reverse engineering. Reverse engineers have typically kept the resulting know-how secret in order to have a competitive advantage over those who have not engaged in this activity.

Publishing information learned through lawful reverse engineering has long been legal in the U.S. In *Chicago Lock Co. v. Fanberg*,³⁷⁷ for example, the Ninth Circuit Court of Appeals overturned an injunction against publication of a book containing key codes for tubular locks whose manufacturer claimed the codes as trade secrets. Because the author of the book, himself a locksmith, had gathered the information from fellow locksmiths who had lawfully reverse-engineered the information in the course of helping their customers, there was no misappropriation of Chicago Lock's trade secrets.

In recent years, restrictions on publication and other disclosures of reverse engineered information have begun to crop up. In the early 1990's, for example, the European Union adopted a directive on the legal protection of computer software that forbade publication or licensing of information obtained in the course of lawful decompilation of programs to achieve interoperability.³⁷⁸ In 1998, the U.S. Congress adopted the DMCA anti-circumvention rules that impose numerous restrictions on disclosure of information learned in the course of privileged acts of reverse engineering. A reverse engineer can, for example, bypass technical protections when necessary to achieve program-to-program interoperability, but cannot disclose information learned therefrom unless the sole purpose of the disclosure is accomplishing interoperability.³⁷⁹ One judge has opined that a journalist's publication of such information would violate the DMCA, even if the information was lawful under the interoperability exception.³⁸⁰ The presentation of a scientific paper on flaws in digital watermarking technology has been challenged as a violation of the DMCA anti-circumvention rules.³⁸¹ Although the DMCA's exception for encryption research permits some dissemination of the results of

³⁷⁷ 676 F.2d 400 (9th Cir. 1981).

³⁷⁸ European Software Directive, *supra* note 177, art. 6(2). The EU rule essentially puts each firm that wants to reverse engineer to the full expense of decompiling the program on its own. This preserves the lead-time of the firm whose program has been decompiled, but leads to more socially wasteful costs unless the decompilee licenses interface information to foreclose the decompilation effort. At least one commentator has opined that publishing reverse-engineered information about the internal design elements of computer software should be illegal. See, e.g., Davidson, *supra* note 162, at 1074-75.

³⁷⁹ 17 U.S.C. sec. 1201(f)(3).

³⁸⁰ *Universal City Studios, Inc. v. Reimerdes*, 111 F.Supp.2d 294, 320 (S.D.N.Y. 2000), *aff'd*, 2001 U.S. App. LEXIS 25330 (2d Cir. 2001). This aspect of the *Reimerdes* ruling is difficult to square this with the Supreme Court's decision in *Bartnicki v. Vopper*, 2001 U.S. Lexis 3815 (U.S. 2001) in which the Court held that a journalist could not be held liable for publishing illegally obtained information as long as the journalist did not participate in the illegal interception of the information.

³⁸¹ See *supra* note 332 and accompanying text. See also Julie E. Cohen, *Unfair Use*, *The New Republic Online*, <http://www.tnr.com/online/cohen052300.html> (discussing Microsoft's claims that online discussion of how to bypass online contract violated the DMCA anti-circumvention rules); John Schwartz, *Apple Offers More Than an Update to Its System*, *N.Y. Times*, Dec. 1, 2001, at C14 (discussing Apple's claim that online posting of information enabling access to a software upgrade violated the DMCA rules).

legitimate encryption research,³⁸² it puts encryption researchers at risk if they publish their results on the Internet because courts might decide this facilitates infringement.³⁸³

When it comes to restrictions on publication, it may be that the economic considerations underlying the DMCA rules are in irreconcilable conflict with values embodied in the first amendment to the Constitution.³⁸⁴ Moreover, economic considerations themselves may be in conflict. Publication of circumvention information, such as the DeCSS software, may have the same market-destructive potential as if its author trafficked in circumvention tools for purposes of facilitating copyright infringement. However, this destructive potential must be weighed against rights of free speech and against another economic purpose, which is to further encryption and computer security research.³⁸⁵

B. Policy Options When Innovators Try to Bypass Reverse Engineering

The very reasons that reverse engineering is socially beneficial—for example, in eroding a first comer’s market power and promoting follow-on innovation—may be why some innovators desire to bypass reverse engineering altogether or render it moot. When reverse engineering is lawful, firms may seek to thwart this activity in one of two ways: by requiring customers to agree not to reverse engineer the product or by designing the product to make it very difficult or impossible to reverse engineer. This subsection addresses the policy responses available to deal with attempts to circumvent legal rules permitting reverse engineering.

1. Avoiding the Threat of Reverse Engineering By Contract

Software licenses often prohibit reverse engineering, even when (or especially when) reverse engineering is allowed by law.³⁸⁶ Whether such contracts should be enforceable as a general matter is an unsettled question of law, as Section IV has shown.

We believe that in markets for products heavily dependent on intellectual property rights, such as computer software, there is reason to worry about contractual restrictions against reverse engineering. Some market power is inevitable in such markets, or else the intellectual property right has no purpose. The policy levers that define the intellectual property are devices that both grant market power and limit its boundaries. If the intellectual property regime is well designed in the first place, we see no intrinsic reason why contracting should be allowed to circumvent it, especially in markets with strong network effects.³⁸⁷ Hence, it may be reasonable not to enforce contract terms purporting

³⁸² 17 U.S.C. sec. 1201(g).

³⁸³ See supra Section V-C.

³⁸⁴ See, e.g., Bellovin Amicus Brief, supra note 328.

³⁸⁵ See supra Section V-C.

³⁸⁶ A parallel policy problem is whether to enforce contractual overrides of fair use and first sale rights of copyright law. See, e.g., Digital Dilemma, supra note 252, at 101-02; McManis, supra note 235.

³⁸⁷ See, e.g., Lemley & McGowan, supra note 47, at 523-27. See also U.S. v. Microsoft Corp., 243 F.3d 34 (D.C. Cir. 2001) (“The company claims an absolute and unfettered right to use its intellectual property as it

to override reverse engineering privileges in intellectual property dependent markets such as software,³⁸⁸ as the European Union has done by nullifying license terms forbidding decompilation of computer programs.³⁸⁹

2. Avoiding the Threat of Reverse Engineering by Technical Obfuscation

Firms sometimes design their products so that it will be difficult or impossible to reverse engineer them.³⁹⁰ Such expenditures would be unnecessary if reverse engineering was unlawful. In the economic calculus about reverse engineering, we must count expenditures to thwart reverse engineering as socially wasteful. Efforts to thwart reverse engineering may, however, be unsuccessful, or only partially successful. Determined second-comers may manage to figure out enough through reverse engineering to make a competitive product, albeit one missing some of the innovator's "secret sauce." Sometimes, however, efforts to circumvent reverse engineering may be successful. In addition, even when firms don't intentionally design their products to make reverse engineering impossible, products may, as a practical matter, be immune from reverse engineering because of the sheer complexity of the product or because details of the product design change so rapidly that by the time reverse engineer finished his work, the next version of the product would be in the marketplace.

One policy option for dealing with such a situation is to force the innovator to disclose certain information about its product.³⁹¹ For example, if the arguments in favor of open interfaces have merit and interfaces cannot be effectively discerned by reverse engineering, then it may sometimes make sense to require interfaces to be made public. This is essentially what happened some years ago in Europe when antitrust authorities brought suit against IBM Corporation for abuse of dominant position because IBM had been altering the interfaces to its mainframe computers frequently, which disadvantaged

wishes...That is no more correct than the proposition that use of one's personal property, such as a baseball bat, cannot give rise to tort liability.")

³⁸⁸ See supra Section IV-C. Of course, contracts that prohibit reverse engineering do not entirely render the reverse-engineering right moot. Assuming the product is available in the market from another source, a potential reverse engineer may have the option to decline the license, and reverse engineer instead. This option will have a salutary impact on the contract terms that are offered, which creates some benefits even if the right to reverse engineer is given up.

³⁸⁹ EU Software Directive, supra note 177, art. 9(1). The nullification extends only to decompilation for purposes of achieving interoperability.

³⁹⁰ See supra note 49.

³⁹¹ It is worth pointing out that in a variety of other circumstances, legal decision-makers have forced firms to disclose information pertaining to publicly distributed products that are not readily discernible from examination of the product when necessary to achieve some important public purpose. While such regulations have sometimes been challenged as unjustified "takings" of private property, for the most part, such challenges have not been successful. See, e.g., *Ruckelshaus v. Monsanto Co.*, 467 U.S. 986 (1984) (challenging requirements of the Federal Insecticide, Fungicide, & Rodenticide Act to submit safety test data to the Environmental Protection Agency which the EPA could consider in connection with a competitor's application for permission to sell the same chemical). The idea of forced disclosure also underlies the proposal of Professors Burk & Cohen for a key escrow system to enable prospective fair users to get access to encryption keys so that they can make fair uses of technically protected digital content. See Dan L. Burk & Julie E. Cohen, *Fair Use Infrastructure for Copyright Management Systems*, Harv. J. L. & Techn. (forthcoming 2001).

European makers of peripheral products. The dispute was eventually resolved by an agreement by which IBM would make advance announcements of changes to its interfaces so that peripheral manufacturers could adjust their products accordingly.³⁹² Some have suggested a similar remedy in *United States v. Microsoft*.³⁹³ Microsoft has maintained its monopoly position in the operating systems market in part through control over the APIs to the Windows platform. Reverse engineering of the Windows APIs is certainly far more difficult than, say, reverse engineering interfaces to game platforms and may be impracticable. Forcing Microsoft to publish its APIs would certainly erode its market power, but this raises a host of other difficulties.³⁹⁴

VII. Conclusion

Reverse engineering is fundamentally directed to discovery and learning. Engineers learn the state of the art not just by reading printed publications, going to technical conferences, and working on projects for their firms, but also by reverse engineering others' products. Learning what has been done before often leads to new products and advances in know-how. Reverse engineering may be a slower and more expensive way for information to percolate through a technical community than patenting or publication, but it is nonetheless an effective source of information.³⁹⁵ Of necessity, reverse engineering is a form of dependent creation, but this does not taint it, for in truth, all innovators stand on the shoulders of both giants and midgets.³⁹⁶ Progress in science and the useful arts is advanced by dissemination of know-how, whether by publication, patenting or reverse engineering.

We think it is no coincidence that most of the proposals to restrict reverse engineering in the past two decades have arisen as to information-based products, such as semiconductors and software. The high quantum of know-how that such products bear

³⁹² See, e.g., *Band & Katoh*, supra note 162, at 22, n. 30. But see *Berkey Photo, Inc. v. Eastman Kodak Co.*, 603 F.2d 263 (2d Cir. 1979), cert. denied, 444 U.S. 1093 (1980) (monopoly firm had no duty under the antitrust laws to predispose information about a new camera and film format to enable competitors in the film market to prepare compatible products).

³⁹³ See, e.g., Piraino, supra note 196, at 888-89. See also R. Craig Romaine & Steven C. Salop, *Slap Their Wrists? Tie Their Hands? Slice Them Into Pieces? Remedies for Monopolization in the Microsoft Case*, 13 *Antitrust* 15, 18 (Summer 1999).

³⁹⁴ A key difficulty arises from the fact that program interfaces are not always self-evident or self-defining. See, e.g., Czarnota & Hart, supra note 177, at 37-38, and *Band & Katoh*, supra note 162, at 6-7 (discussing difficulties of precisely defining "interface"). Much judicial oversight might be necessary to enforce an obligation by Microsoft to disclose interface information. See Romaine & Salop, supra note 393, at 19.

³⁹⁵ As Dreyfuss & Kwall observe, "Since there is no time limit to trade secrecy protection, reverse engineering is the principal way in which a trade secret enters the public domain." Dreyfuss & Kwall, supra note 50, at 818.

³⁹⁶ That is, progress happens through both breakthrough innovations and the accumulation of small steps. Economists have focused on designing standards of patentability and breadth in order to balance the incentives of earlier and later innovators. See, e.g., Scotchmer, supra note 104 (focusing on breakthrough inventions), O'Donoghue et al, supra note 140 (focusing on incremental innovation), and Nancy T. Gallini & Suzanne Scotchmer, *Intellectual Property: When Is It the Best Incentive?* in 2 *INNOVATION POLICY AND THE ECONOMY* (Adam Jaffe, Joshua Lerner, & Scott Stern, eds. 2001). Some legal scholars have proposed supplementary legal regimes to deal with subpatentable innovations. See, e.g., Reichman, *Legal Hybrids*, supra note 14; Manifesto, supra note 15.

on or near the face of the product make these products more vulnerable than traditional manufactured goods to market-destructive appropriations.³⁹⁷ This is especially true when the information is in digital form. Copying and distribution of digital products is essentially costless and almost instantaneous in the digital networked environment. The vulnerability of information products to market-destructive appropriations may justify some limitations on reverse engineering or post-reverse engineering activities, but reverse engineering is important to innovation and competition in all industrial contexts studied.

Adapting intellectual property law so that it provides adequate, but not excessive, protection to innovations is a challenging task. In considering future proposals to limit reverse engineering, we hope that policymakers will find it helpful to consider the economic effects of mechanisms that have been employed in the past. Restrictions on reverse engineering ought to be imposed only if justified in terms of the specific characteristics of the industry, a specific threat to that industry, and the economic effects of the restriction.

We worry that the recent DMCA restrictions on reverse engineering may propagate backwards and erode longstanding rules permitting reverse engineering in other legal regimes. As Professors Dreyfuss and Kwall have observed, “the distinction between, say, breaking into a factory (improper) and breaking into the product (proper) may seem artificial.”³⁹⁸ It is, however, a distinction that has been a foundational principle of intellectual property and unfair competition law, at least until enactment of the DCMA. It is, moreover, a distinction whose abandonment could have detrimental consequences for innovation and competition.

³⁹⁷ See, e.g., Reichman, *Legal Hybrids*, supra note 14, 2443-44.

³⁹⁸ Dreyfuss & Kwall, supra note 50, at 818.