The Cryptic Case for Public-Key Patents

by Julianne Nelson

Columbia Institute for Tele-Information
Graduate School of Business
Columbia University
809 Uris Hall
New York, NY 10027
(212)854-4222

The Cryptic Case for Public-Key Patents

Julianne Nelson (jnelson@american.edu)
Assistant Professor, School of Public Affairs
American University
4400 Massachusetts Avenue, N.W.
Washington D. C. 20016

The early 1990s provide evidence of dramatic changes in US patent policy, including the review and revocation of several highly controversial patents,[1] a new GATT treaty that changed both the length and character of patent duration,[2] a successful patent-infringement claim against Microsoft for $120 million,[3] and a tacit promise by the federal judiciary to clarify the "doctrine of equivalents" in U.S. patent law.[4] The GATT treaty and the jury verdict against Microsoft clearly have the

[1] In *The New York Times* of December 8, 1994 ("U.S. Revokes Cotton Patents After Outcry From Industry," Section D, page 1), Teresa Riordan reported that in less than one year, Bruce Lehman (the Commissioner of Patents and Trademarks) had cancelled four controversial patents: two previously awarded to Agracetus for genetically engineered cotton, one previously awarded to Compton's NewMedia Inc. for CD-ROM search techniques, and one patent previously awarded to Software Advertising Corporation (SAC) for a method of using advertising logos in software display screens. **(Appeals status)**

[2] In *The New York Times* of January 16, 1995 ("Patents," Section D, Page 2), Susan Chartrand reported that after June 8, 1995, patents will be valid for 20 years from the date of application (rather than 17 years from the date of issue). This change was included in the GATT agreement passed by Congress and signed by President Clinton in December 1994. For the transition period, a patent awarded on the basis of an application filed *before* June 8, 1995 (but expiring after that date) will last the maximum of 17 years from the date of issue or 20 years from the date of appliction. Chartrand also reported that the Patent and Trademark Office planned to introduce a new "provisional" patent (a *temporary* patent designed to protect an inventor for up to one year before the official life of the regular patent begins) and to allow a three-year extension of the 20 year patent for circumstances beyond the control of the inventor. **(Legislation to extend the interim rule?)**

[3] In "Do Software Patents 'Stac' the Deck Against the Competition?" (*Computer Shopper*, Vol. 11, No. 4, (April 1994) p. 1), James Morando and Christian Nadan reported that Stac was awarded approximately $5.5 for each sale of DOS 6.0 and 6.2 -- the Microsoft operating systems containing the compression program that infringed Stac's patent.

[4] On December 3, 1993, the U.S. Court of Appeals for the Federal Circuit agreed, in its review of *Hilton Davis Chemical v. Warner-Jenkinson*, to hear arguments concerning the nature and extent of the proof needed to justify a finding of patent infringement under the doctrine of equivalents. The rule in force prior to this case required proof that the allegedly infringing invention performed "substantially the same function in substantially the same way to give substantially the same result." (*Graver Tank & Mfg. Co. v. Linde Air Products Co.*, 339 U.S. 605, 608 (1950)). The decision in *Hilton Davis* is anticipated in early 1995.

potential to strengthen patent protection, while the activism of the new Commissioner of Patents and Trademarks has the potential to undermine it. The future implcations of the doctrine of equivalents remains unsettled. The diverse nature of these developments leaves me wondering which (if any) were good for specific industries and for the economy as a whole. This paper represents an initial attempt to satisfy my curiosity on this issue as it concerns public-key cryptography.

1 *Public-Key Cryptography and the Debate over Software Patents*

A cursory glance at the trade press reveals a long-running shouting match between those generally favoring and those opposing software patents. Each side of the debate can produce endorsements from both academics and practitioners. Supporters of software patents include the author of a major patent law treatise Donald Chisum,[5] software giants like Borland, Microsoft, Intel, and Apple,[6] employees of hardware manufacturers like Carl Dichter of Motorola[7], and a great many small software developers (like Paul Heckel[8] and others who testified at hearings held in 1994 by the

[5] See "The Future of Software Protection: The Patentability of Algorithms." (47 *U. Pitt L. Rev.* 959) for Chisum's argument that the Supreme Court should overrule its 1972 decision in *Gotschalk v. Benson* that limited the patentability of mathematical algorithms.

[6] John Gliedman, "Software patent row divides industry," *Computer Shopper*, Vol. 14, No. 7 (June 1994), p. 61.

[7] See "Patently Wrong: Software Patents," (*UNIX Review*, Vol 10, No. 11 (November 1992), p. 38) for Dichter's argument that patents provide the incentive needed to encourage improvements on basic inventions.

[8] In "Debunking the Software Patent Myths," (*Communications of the ACM*, Vol. 35, no. 6 (June 1992), p. 121), Heckel responds to the charge made by the League for Programming Freedom (in the January 1992 issue of *Communications*) that his patent #4,736,308 (along with eight others) was "absurd." When the article was written, Paul Heckel owned a small software company, and worked as a consultant on intellectual property law for computer software.

PTO[9]). Opponents of software patents include law professor Paula Samuelson[10], computer science professor Allen Newell[11], computer companies Adobe, Autodesk, and Oracle,[12] the editor-in-chief of *P.C. Magazine* Michael J. Miller[13], various Cypherpunks[14] and the members of Richard Stallman's League for Programming Freedom.[15]

Do the criticisms of software patents apply to public-key cryptography patents? One is tempted to label the latter as "hardware patents" and thereby to exempt them from the attacks addressed to software patents. The first two public-key patents did not include the word software, but were instead written for a "cryptographic apparatus"[16]; the third public-key patent was for a

[9] See Gregory Aharonian's review of the February 10 hearings posted at gopher cite patents@world.std.com.

[10] See "Should Program Algorithms Be Patented?" (*Communications of the ACM*, Vol. 33, No. 8 (August 1990), p. 23) and "Benson Revisited: The Case Against Patent Protection for Algorithms and Other Computer-Related Inventions," (39 *Emory Law Journal* 1025 (1990)) for her review of the case law that supports limitations on the patentability of algorithms.

[11] See "Response: The Models Are Broken!", (47 *U. Pitt. L. Rev.* 1023 (1986)) for his argument that algorithms are indistinguishable from "mental processes" and should therefore be unpatentable.

[12] Gliedman (1994, p. 61).

[13] See "Software Patents Must Go." (*PC Magazine*, Vol. 13, No. 5 (March 15, 1994), p. 79) for his argument that the laws of copyright and trade secrecy should suffice to protect the intellectual property of software developers.

[14] See Steve Levy, "The Battle of the Clipper Chip", *New York Times*, Section 6, p. 46, for description of the organization co-founded in September 1992 by Erric Hughes and Tim May.

[15] See "Against Software Patents," *Communications of the ACM* (Jan. 1992) and "Software Patents," *Dr. Dobbs Journal* (November 1990) for a statement of the League's position on this issue.

[16] In particular, see the descriptions of the "Stanford patents": patent #4,200,770 awarded to Diffie, Hellman and Merkle on April 29, 1980 and patent #4,218,582 awarded to Hellman and Markle on August 19, 1980. The choice of wording is hardly surprising since machines at the time were too slow to run an equivalent software implementation of public key cryptography.

"cryptographic communications system and method" using encoding and unencoding devices.[17] More than a decade later, practitioners in 1990 were still recommending hardware implementations of public-key encryption over their software equivalent.[18] In 1994, the Clinton administration sought to adopt a form of hardware cryptography -- the Clipper chip -- as a govenment-sanctioned standard.[19] If public-key cryptography were readily distinguishable from software, then it would be easier to dismiss one set of potential critics and to find strong support for public key cryptography in the general acceptance of hardware patents.

Unfortunately, this simple solution does not withstand close scrutiny. Many forms of software encryption have been "facilitated" by hardware for years.[20] Recent advances in technology have now made it possible to implement the public key patents as software.[21,22] As the technology continues to evolve, distinctions between encryption hardware and software will grow increasingly artificial. These labels will not provide a meaningful basis for determining what should and should not be patented. It follows that the supporters of cryptography patents cannot easily dismiss the

[17] See the description of patent #4,405,829 awarded to Rivest, Shamir and Adelman (the RSA patent) on September 20, 1983 (the "RSA" patent).

[18] Keith Jackson, *Secure Information Transfer, PC Encryption: A Practical Guide*, Boca Raton, Fla.: CRC Press, 1990, pp. 32-33.

[19] Get details of the approval and revocation of the standard.

[20] For example, Jackson (1990, p. 32) reports that the CryptoGard "blurred the distinction between hardware and software encryption" by providing "a software implementation of the DES algorithm using the processor on the card."

[21] In February 1995, a search of patents on record revealed 56 filings that cited both the word "software" and at least one of the controlling public key cryptography patents.

[22] Cite statistics on the industry-wide shift in innovations from hardware to software mentioned in Burke.

arguments made by the critics of software patents.

2 *The Economic Rationale(s) for Patents*

Economists generally take the existence of some patent protection as a given and proceed to derive the optimal version of the policy. They rely on models that typically specifies the relevant set of players (i.e., would-be inventors, product developers, and consumers, all assumed to pursue a well-defined set of objectives), an endowment for each player (i.e., a description of the options available to each player in every state of the world in each period), and the set of feasible trades.[23] A description of the available markets (and their transaction costs, if any) makes it possible to define the equilibrium supported by a given set of assumptions. For each individual in a given equilibrium, the marginal private benefit of each discretionary activity (like research and development effort) will just equal its marginal private cost.[24] Ideally the marginal *social* benefit of each such activity will also equal its marginal *social* cost.

Although this methodology represents a common ground shared among economists, an initial review of the literature yields a diversity of opinion about efficient patent policy that echoes the industry debates over software patents. Some economists argue that patents give inventors a much-needed jackpot to strive for[25], while others worry about "patent races" and wasteful, duplicative

[23] Cite to a finance text that describes an economy with "complete markets."

[24] This result follows as an interpretation of the first order necessary conditions for an extremum. If each individual maximizes net benefit defined as B(e) - C(e), then a necessary condition for a solution to this optimization problem is B'(e) - C'(e) = 0.

[25] Arrow, Nordhaus, Scherer.

research expenditures.[26] Some economists recommend giving broad patent protection to industry "pioneers" in order to encourage them to develop their new propsects,[27] while others worry that broad pioneer patents unduly hinder research on "cumulative technologies."[28]

These differences in policy recommendations can generally be traced to different basic assumptions about the nature of the industry in question. In recent work, Robert Mazzoleni and Richard Nelson have identified four general purposes to be served by patent awards, with each purpose having its distinct implications for optimal policy:

P1. Patents provide motivation for invention.

P2. Patents induce inventors to "disclose" their inventions when they would otherwise rely on secrecy for protection.

P3. Patents induce the development of inventions.

P4. Patents enable the orderly development of prospects.[29]

Mazzoleni and Nelson argue that patents best serve purpose P1 in industries where diverse inventors pursue non-competing innovations for the sake of the profits to be earned from using the patented

[26] In section 2 of "The Timing of Innovation: Research, Development, and Diffusion" (*Handbook of Industrial Organization, Volume I*, R. Schmalensee and R. Willig eds., Elsevier Science Publishers, 1989), J. Reinganum reviews the argument that "aggregate expenditure on R&D is too high relative to the cooperative optimum; there are too many firms and each invests too much."

[27] See Kitch, "The Nature and Function of the Patent System" (20 *Journal of Law and Economics* 265 (1977)) for a version of this argument.

[28] See R. Merges and R. Nelson, "On the Complex Economics of Patent Scope" (90 *Columbia Law Review* 839, 908-909 (May 1990)) for a version of this argument.

[29] R. Mazzoleni and R. Nelson, "Economic Theories about the Benefits and Costs of Patents," Working Paper, 1995, p. 2.

technology.[30] They argue that in order for patents to serve purpose P2, trade secrecy must be a *viable* alternative to disclosure through patent application. Patents may serve either purpose P3 or P4 when technology evolves through a series of stages (i.e., innovations are cumulative). It remains to determine which purposes, if any, are served by patents on public-key cryptography.

3 *Cryptography Patents as Incentives for Innovation*

Economists generally assume that entrepreneurs are primarily motivated by profit and that productive effort reqires personal sacrifices. If the creative effort needed to produce useful inventions is costly and discretionary, then it is plausible to argue that increasing the value of patents (by, for example, increasing their scope or duration) will increase the number of new inventions. However, evidence suggests that the need for a profit does not provide a strong justification for cryptography patents.

Table 1, based on empirical work by Mansfield,[31] indicates that patents provide stronger incentives for innovation in some industries than in others. If cryptographic devices are like electrical equipment generally during this period, then 96 percent of patented inventions would be ~~have been~~ introduced even without the protection of intellectual property law. This view was echoed in 1981 by Intel Chief Counsel Roger Borovoy when he was quoted as saying

> "In the electronics industry, patents are of no value whatsoever in spurring research and development. We use them because we have to. You can't be the only holdout against the

[30] In contrast, if inventors are working on competing innovations, the patent system may actually encourage too many innovations.

[31] Edwin Mansfield, "Patents and Innovation: An Empirical Study." *Management Science*, Vol. 32, 1986, pp. 173-181.

angry hordes or else you pay everyone.[32]

Industry	Percent of Products that Would *Not* Have Been Introduced (1981-83)	Percent of Products that Would *Not* Have Been Developed (1981-1983)
Pharmaceuticals	65	60
Chemicals	30	38
Petroleum	18	25
Machinery	15	17
Fabricated Metal Products	12	12
Primary Metals	8	1
Electrical Equipment	4	11
Instruments	1	1
Office Equipment	0	0

Arguments offered by those on both sides of the debate over software patents also tend to support my contention that innovation provides a weak justification for cryptography patents. Menell, a skeptic when it comes to software patents, observes that

> government and private subsidies of research, expecially at universities which publish and otherwise freely disseminate their discoveries, alleviate the public goods[33] problem. Many

[32] "The Patent is Expiring as a Spur to Innovation," *Business Week*, May 11, 1981, p. 44E, as reported in Burke (1994, p. 1132).

[33] i.e., goods like national defense, broadcast television, and information generally that are "nonexcludeable" and "nonrival": no one can readily be prevented from consuming the good and the cost of an additional consumer is arbitrarily small. Economic theory predicts that the market will provide too few of such goods (in the absence of government intervention) because individual

> ground-breaking areas of computer software research have been supported generously in this manner.[34]

When Thomas Burke argues *in support* of software patents, he acknowledges

> that universities have been heavily dependent on substantial government subsidies from such sources as the Department of Defense's Defense Advanced Research Projects Agency (DARPA) and NASA. A common pattern has been that computer scientists conduct their pioneering work at universities and then proceed to attract the venture capital with their own companies once they have a product that is commercially viable.[35]

If these characterizations of the software industry is accurate, development incentives provide a stronger justification for software patents than do research incentives.

The specific history of public-key cryptography suggests that goals other than profit -- like professional status, tenure, satisfied curiosity, or even the chance to disrupt government security agencies -- influenced industry innovators in the past. Much of the early work in public-key cryptography fit the scenario described by Menell and Burke: the research was done by university researchers and financed in part by the government.[36] Recent reports in *Bussiness Week* indicate that RSA Data Security Inc. -- the company founded by the inventors of the RSA patent -- has not yet fully succumbed to the lure of profits. Russell Mitchell reports that "even though it's in a position to do so, it is not gouging on prices."[37]

investors cannot realize the full marginal social benefit of their productive efforts.

[34] "The Challenges of Reforminig Intellectual Property Protection for Computer Software," 94 *Columbia Law Rev.* 2644, 2647.

[35] Burke (1994, p. 1125, note 45).

[36] Both of the "Stanford patents" were financed in part by NSF grant ENG-10173 and IPA No. 0005. The RSA patent was financed in part by Department of the Navy Contract N00014-67-A-0204 and by NSF grant MCS76-14249.

[37] "The Key to Safe Business on the Net," *Business Week*, Feb. 27, 1995

Philip Zimmermann -- a Cypherpunk and the author of PGP -- provides an example of a different motivation at work. He has been described as an "electronic-freedom fighter" who wrote the public-key encryption PGP program to " 'innoculate the body politic' form the danger of government prying."[38] Or as Levy (1994) explains it,

> ...Zimmerman hoped that the appearance of free cryptography would guarantee its continued use after a possible Government ban. One of the first people receiving the program [in 1991] placed it on a computer attached to the Internet and within days thousands of people had PGP.[39]

It seems safe to say that considerable creative activity would have occurred in this industry even in the absence of patent protection.

4 *Cryptography Patents and Trade Secrecy*

A variety of authors cite disclosure incentives as a benefit of patent protection. For example, Burke (1994, p. 1132) quotes the justification for patents found in the U.S. Code: "the patent system fosters the dispersal of knowledge by requiring patent applicants to reveal the 'best mode' for practicing their inventions."[40] Mazzoleni and Nelson (1995, pp. 6-7) suggest that ascribing this benefit to patents presupposes that trade secrecy is feasible, i.e., that reverse engineering is relatively difficult. If reverse engineering were relatively easy, then the patent system would have little effect. Any innovation could be readily discovered whether it was reported publicly or not.

[38] William Bulkeley, "Cipher Probe: Popularity Overseas of Encryption Code has the U.S. Worried" (*Wall Street Journal*, Vol. CCXXIII, No. 83 (Thursday, April 28, 1994), p. 1).

[39] Mention innovation embodied in PGP -- the choice of random number for a seed; the allegations of infringement made by RSA and the final (?) resolution of the feud with the creation of PGP 2.6, a fully-licensed version.

[40] 36 U.S.C. section 112 paragraph 1 (1988).

The history of public-key cryptography suggests that trade secrecy is quite difficult to maintain and that voluntary disclosure would often occurred even without the existence of patents. Concerning the second of these two claims, the innovation patented by Diffie, Hellman and Merkle was publicly announced in June 1976 at the National Computer Conference.[41] The inventors did not seem to be in a hurry to file for a patent after their discovery was announced. Since the patent application was not filed until September 6, 1977, more than a year after the initial public disclosure, there is even an argument that the patent invalid and unenforceable.[42]

A recent incident involving public-key cryptography software suggests that reverse engineering (or other means of obtaining trade secrets) is relatively easy in this industry. According to *Informaton Week* of October 3, 1994, "The source code for RSA's RC4 program was posted from a home computer in Texas by an unidentified user, in apparent violation of non-disclosure, trade secret, and export laws." [43] The secrecy of this source code had formed the basis for an agreement between the National Security Agency and the Software Publisher's Association: the cryptography algorithm it contained was ~~the only one~~ one of the few easily approved for export.[44] The fact that RSA could not prevent information this important from being discovered casts serious doubt on the wisdom of

[41] This fact appears in allegations 36-38 in a complaint filed by Roger Schlafly against PKP and RSA Data Security, Inc. (Civil Action File C-94 20512 for the U.S. District Court for the Northern District of California, hereinafter referred to as the "Schlafly complaint"). Schlafly cites W. Diffie, "The First Ten Years of Public-Key Cryptography", (*Proceedings of the IEEE*, vol. 76, no. 5, May 1988) as his source of information.

[42] *ibid.*

[43] It is not known whether or not the code posted was, in fact, the actual source from a licensed use of the algorithm, such as the mass-market program "Lotus Notes."

[44] John Markoff, "A Secret Computer Code is Out," *New York Times*, Sept. 17, 1994, Section 1, p. 37.

relying on trade secrecy in this industry, period.

5 *Cryptography Patents and Product Development*

The evidence reviewed thus far indicates weak support for cryptography patents. However, there is a stronger case to be made for these patents from the product development ~~incentives~~ opportunities they provide. To begin, Mansfield's early empirical work reported in Table 1 suggests that product development was more sensitive than basic research to the existence of electrical equipment patents. The same can be said for the anecdotal evidence found in the debate over software patents: a wide variety of authors emphasized how patents made it possible for companies to bring new products to market.

Aspects of the current structure of the market for public-key cryptography provide further support for this hypothesis. Revenues from the licensing of patents and software toolkits have enabled RSA Data Security, Inc. to undertake a variety of educational activities at a rate probably beyond that consistent with strict profit-maximization. For example, the company

organizes and underwrites an annual cryptography conference (thereby providing industry participants with the chance to discuss recent developments);

undertook an active campaign to sink the clipper chip (thereby making it possible for firms to continue using public-key encryption techniques[45]);

subsidizes a "shareware promoter" (Consensus Development) to help would-be entrepreneurs develop their applications of the RSA patents;

has structured licensing fees to encourage small software developers: the commercial license

[45] Levy (1994) quotes James Bidzos, the president of RSA, as saying "For almost 10 years I've been going toe to toe with these people at Fort Meade. The success of this company is the worst thing that could happen to them. to them, we're the real enemy."

for RSAREF has a low "start-up" fee (but a higher percentage royalty rate);

has released some of its algorithms into the public domain (the hash algorithms MD4 and MD5 are available for unrestricted use[46])

has made patented innovations freely available for non-commercial use since 1989; and

has established a research and consulting division (RSA Laboratories);

Nevertheless, these claims alone do not enable us to evaluate the impact of patents on public-key cryptography. It remains to determine whether or not the current scope of cryptography patents provide too much development incentive or too little.

does mb=mc provide a complete specification of patent policy?

implications of Hart & Moore (JPE: 12/90) --implementation of Nash cooperative bargaining solution as a means of defining an "optimal assignment of property rights"

examples:

bargain between Cylink & RSA (to form PKP);

between RSA and Zimmerman(?);

between RSA & generic developer.

Question: how should the joint surplus from the innovations be split?

[46] Paul Fahn. "Answers to Frequently Asked Questions About Cryptography Today," Version 2.0 (Sept. 20, 1993), Section 8.3.