CYBER VIEW

Cracking the U.S. Code

Late last winter a graduate student at the University of California at Berkeley needed only three and a half hours to crack a message encoded in the strongest legally exportable cipher in the U.S. He used the spare processing cycles of a few hundred workstations on the campus network. Although computer scientists and high-tech companies all agree that more secure codes should be widely used, the U.S. government continues to come down hard on would-be purveyors of cryptography. And courts in California and Washington, D.C., have issued diametrically opposed opinions about the legitimacy of government controls over cryptographic software.

It has been almost five years since Daniel J. Bernstein, now a professor at the University of Illinois, first asked the State Department whether he could be jailed for distributing a technical paper on cryptography and two pages of program code illustrating the results of his research. He has yet to receive a straight answer.

The point of Bernstein's paper was to demonstrate that some innocuous-looking and widely used mathematical functions could encrypt files as well as more obviously dangerous algorithms. When he first asked the State Department for separate rulings on the paper and the programs, the Bureau of Politico-Military Affairs claimed that the paper served as documentation for the programs. So they combined the requests and denied them both, citing the International Traffic in Arms Regulations (ITAR), which govern publication of cryptographic information. But in mid-December federal Judge Marilyn Patel ruled that ITAR was a classic example of unconstitutional restraint on free speech and that Bernstein could not be prosecuted under them. At the end of the month, the Clinton administration issued new regulations that transfer jurisdiction to the Commerce Department but otherwise could subject Bernstein and anyone else who teaches or writes practical information about cryptography to heavy fines or jail terms.

The new regulations also contain a peculiar clause that forbids bureaucrats deciding whether to grant an export license for an encryption system from taking into account whether equivalent or identical software is already available overseas. Software firms and individuals such as Bernstein had previously tried to bolster their cases with lists of the nearly 2,000 strong-encryption software packages available outside the U.S.

About the time that Bernstein's travails were beginning, Bruce Schneier authored a book entitled *Applied Cryptography,* which discusses many commonly used ciphers and included source code for a number of algorithms. The State Department decided that the book was freely exportable because it had been openly published but refused permission for export of a floppy disk containing the same source code printed in the book. The book's appendices on disk are apparently munitions legally indistinguishable from a cluster bomb or laser-guided missile. In early 1996 federal Judge Charles R. Richey dismissed the lawsuit to overturn this decision, brought by Schneier's collaborator, Philip R. Karn, Jr. Richey cited among other things a clause in ITAR that exempts decisions under them from judicial review.

DAVID SUTER

The new regulations do not contain the exemption from review (which Patel had declared unconstitutional). As a result, in January an appeals court in Washington returned Karn's case to Richey, who will determine whether the other reasons he gave for dismissing the case still hold. In the meantime, Karn's disk cannot legally leave the country, even though the original book has long since passed overseas and all the code in it is available on the Internet.

At the heart of both cases is the argument over whether software is a text or a machine. Bernstein and Karn argue that their right to free speech is being violated, but government lawyers contend that the regulations simply prohibit the export of dangerous equipment for concealing information. On the one hand, programs—even in the form of 1's and 0's—can be protected by copyright like other texts. On the other hand—even when described in plain English—they can be patented like other machines. And many computer scientists agree that the best way to explain how a computer program works is simply to give people the code to study.

Advances in computer science are not clarifying matters either. Automatic-programming systems, which transform abstract mathematical specifications into working code, could generate encryption programs from high-level descriptions, says Alan Goldberg, a researcher at the Kestrel Institute in Palo Alto, Calif. (The basic recipe for the strongest public-key cryptographic systems, for example, is: "Treat the characters in the message as digits in a very large number, raise that number to a power, divide it by another very large number and output the remainder.")

Future generations of automatic-programming software, Goldberg says, might even be able to take the basic requirements of cryptography—such as the fact that each bit of information in the input is spread throughout the entire encrypted message—and apply a series of expansions and transformations that would ultimately result in working programs. It would be difficult for the government to argue that such general instructions are readily distinguishable from ordinary speech.

No amount of logic chopping will lead lawmakers out of this dilemma, says Randall Davis of the Massachusetts Institute of Technology. He contends that the fundamental premise of arguments over software's status is flawed because it is both text and mechanism. Any rules based on the notion that these two categories are distinct must eventually come to an impasse, whether they deal with patents, copyrights, munitions or the First Amendment. To date, Davis has little in the way of a grand synthesis between the two apparently incompatible classifications, but it seems clear that something is needed soon before the thus far irresistible tide of software innovation strikes the immovable wall that is the law. *—Paul Wallich*