DANIEL J. BERNSTEIN
Department of Mathematics, Statistics, and Computer Science
University of Illinois at Chicago
Mail Code 249
Science and Engineering Offices, Room 322
851 S. Morgan Street
Chicago, IL 60607–7045
(312) 996–3041
Best address: djb-legal@cr.yp.to

Plaintiff *Pro Se*

IN THE UNITED STATES DISTRICT COURT
FOR THE NORTHERN DISTRICT OF CALIFORNIA

DANIEL J. BERNSTEIN, Plaintiff, v. UNITED STATES DEPARTMENT OF COMMERCE, et al., Defendants.	C 95–00582 MHP **DECLARATION OF DANIEL J. BERNSTEIN RE CRYPTOGRAPHY** Date: October 7, 2002 Time: 2:00 p.m. Place: Courtroom 15, 18th Floor

I, DANIEL J. BERNSTEIN, hereby declare:

1. I am the plaintiff in the above-entitled action. I currently reside in Berkeley, California. Except as expressly stated below, I have personal knowledge of the facts stated herein. If called upon to testify, I would competently testify to these facts.

Impact of the Regulations

2. Before 1999, I severely limited the time I spent on cryptographic research and education. I knew from firsthand experience that working in this field led to legal problems.

3. My time is now somewhat less skewed. However, out of fear of the regulations, I am continuing to limit the time I spend on cryptographic research and education.

4. The defendants appear to believe that EAR's entire effect on me is that my web pages do not contain certain documents that I have filed as exhibits to my declarations. There are three big reasons that this belief is, as a factual matter, incorrect.

5. First, the documents that I have filed with the Court are merely a few examples of my work, meant to illustrate the impact of the regulations. There are many other documents that I have written and that I would like to publish in violation of EAR. I do not want to spend the time necessary to file and explain these documents: discovery is on hold, and I believe that the examples that I have provided are more than adequate to support my motion for summary judgment.

6. Second, there are many other documents that I would like to write and publish in violation of EAR. There are limits to the amount of time that I can afford to spend writing documents without knowing that their publication will be protected by the First Amendment. Obviously I cannot file documents that have not yet been written.

7. Third, publication is only one part of the scientific process. I would also like to engage in prohibited "technical assistance" and "export" of "software" and "technology" in private email and in face-to-face discussions. I do not want to disclose private email messages, and obviously I cannot predict, let alone file, the contents of spontaneous discussions at future conferences.

Examples: snuffle.c et al.

8. My first declaration in this case (docket no. 63) described Snuffle (docket no. 5), and in particular snuffle.c. When snuffle.c is combined with Ralph Merkle's Snefru hash function, it can trivially be used to protect messages against eavesdropping.

9. My declaration also briefly described SEOC, an outgrowth of Snuffle. My second declaration (docket no. 87) mentioned dh227, which includes SEOC. I have used dh227 for years to protect messages against eavesdropping and forgery.

10. I would like to put the collection of snuffle.c and Snefru on my web pages without government notification. I would also like to put dh227 and other similar items on my

web pages without government notification. I would also like to show similar items to my colleagues in private email and in face-to-face discussions without government notification.

11. The collection of snuffle.c and Snefru is, according to the plain meaning of EAR, an "encryption item" controlled by 5D002 for "EI reasons." It uses, and is designed to use, digital "cryptography" (specifically, digital transformation of information using a secret key in order to hide its content) to perform a "cryptographic function other than authentication or digital signature" (specifically, encryption), using a "symmetric algorithm" with a key length above 56 bits. The same comment applies to dh227.

Examples: SPRAY et al.

12. I have written, and would like to put on my web pages without government notification, the collection of SPRAY (docket no. 187, Exhibit E), spray-key.c, spray-add.c, spray-sub.c, and spray-make. I would also like to show similar items to my colleagues in private email and in face-to-face discussions without government notification. Exhibit A is a true and correct copy of spray-key.c. Exhibit B is a true and correct copy of spray-add.c. Exhibit C is a true and correct copy of spray-sub.c. Exhibit D is a true and correct copy of spray-make.

13. This collection is, according to the plain meaning of EAR, an "encryption item" controlled by 5D002 for "EI reasons." It uses, and is designed to use, digital "cryptography" (specifically, digital transformation of information using a secret key in order to hide its content) to perform a "cryptographic function other than authentication or digital signature" (specifically, encryption), using a "symmetric algorithm" with a key length above 56 bits (specifically, 512 bits).

14. A sender and receiver can prepare to use this collection as follows. The sender and receiver meet in person at a secure location. The sender types "`sh spray-make`" on his computer, and the receiver types "`sh spray-make`" on his computer. The sender then types "`spray-key SecretKey`" and bangs randomly on the keyboard for a few minutes. When he is done, he types Enter and Ctrl-D. The sender then gives the receiver a copy of the

resulting `SecretKey` file. (I am assuming that the sender and receiver are both using the Linux operating system.)

15. Later, the sender can send a secret file, such as `MedicalData`, to the receiver as follows. The sender types "`spray-add SecretKey < MedicalData > MedicalDataScrambled`" and then sends `MedicalDataScrambled` to the receiver through email. The receiver saves `MedicalDataScrambled` and types "`spray-sub SecretKey < MedicalDataScrambled > MedicalData`" to recover `MedicalData`. The sender and receiver can exchange any number of secret files this way, without meeting in person again.

16. I designed SPRAY with the following intent: an eavesdropper cannot learn anything about `MedicalData`, other than its length, from the contents of `MedicalDataScrambled`. The other pieces of software—spray-key.c, spray-add.c, spray-sub.c, spray-make—are trivial wrappers that rely on the cryptographic strength of SPRAY.

17. Pseudorandom number generators other than SPRAY can easily be used in the same way. For example, my Introduction to Cryptography (docket no. 187, Exhibit H) includes, among other things, assembly-language software for another pseudorandom number generator, the "Tiny Encryption Algorithm block cipher"; I could easily scramble data using that generator instead of SPRAY.

Examples: nistp224 et al.

18. I have written, and would like to put on my web pages without government notification, the collection of nistp224, SPRAY, s224-key.c, s224-add.c, s224-sub.c, and s224-make. I would also like to show similar items to my colleagues in private email and in face-to-face discussions without government notification. Exhibit E is a true and correct copy of s224-key.c. Exhibit F is a true and correct copy of s224-add.c. Exhibit G is a true and correct copy of s224-sub.c. Exhibit H is a true and correct copy of s224-make.

19. This collection is, according to the plain meaning of EAR, an "encryption item" controlled by 5D002 for "EI reasons." It uses, and is designed to use, digital "cryptography" (specifically, digital transformation of information using a secret key in order to hide its content) to perform a "cryptographic function other than authentication or digital signature" (specifically, encryption), using a "symmetric algorithm" with a key length above 56 bits (specifically, 512 bits) and an "asymmetric algorithm" with a group size above 112 bits (specifically, 224 bits).

20. A sender and receiver can prepare to use this collection as follows. The sender types "`sh s224-make; s224-key SenderSecret SenderPublic`" and bangs randomly on the keyboard. Meanwhile, the receiver types "`sh s224-make; s224-key ReceiverSecret ReceiverPublic`" and bangs randomly on the keyboard. The sender and receiver then exchange their `SenderPublic` and `ReceiverPublic` files. They can do this in public: the contents of those files do not have to be kept secret.

21. Later, the sender can send a secret file, such as `MedicalData`, to the receiver as follows. The sender types "`s224-add SenderSecret ReceiverPublic < MedicalData > MedicalDataScrambled`" and then sends `MedicalDataScrambled` to the receiver through email. The receiver saves `MedicalDataScrambled` and types "`s224-sub ReceiverSecret SenderPublic < MedicalDataScrambled > MedicalData`" to recover `MedicalData`. The sender and receiver can exchange any number of secret files this way.

Cryptographic Strength

22. There are several well-known mathematical theorems demonstrating that various methods of protecting messages against eavesdropping are as strong as the underlying pseudorandom number generators and key-exchange systems. The only hope for the cryptanalyst is to find (1) a cryptographic weakness in the pseudorandom number generator or (2) a cryptographic weakness in the key-exchange system. These theorems are proven by

mathematical analysis of all possible cryptanalytic techniques, not merely the specific techniques known today.

23. An eavesdropper who demands copies of cryptographic software, *except* for pseudorandom number generators and key-exchange systems, will not obtain a complete picture of how messages are being scrambled. In fact, he will obtain essentially none of the picture.

I declare under penalty of perjury under the laws of the United States that the foregoing is true and correct and that this declaration was executed on this 3rd day of September, 2002.

DANIEL J. BERNSTEIN