CINDY A. COHN, ESQ.; SBN 145997
McGLASHAN & SARRAIL
Professional Corporation
177 Bovet Road, Sixth Floor
San Mateo, CA 94402
Tel: (415) 341-2585
Fax: (415) 341-1395

LEE TIEN, ESQ.; SBN 148216
1452 Curtis Street
Berkeley, CA 94702
Tel: (510) 525-0817

Attorneys for Plaintiff
Daniel J. Bernstein

IN THE UNITED STATES DISTRICT COURT

FOR THE NORTHERN DISTRICT OF CALIFORNIA

DANIEL J. BERNSTEIN)
) C 95-00582 MHP
Plaintiff,)
) DECLARATION OF
v.) MICHAEL PAUL JOHNSON
)
)
UNITED STATES DEPARTMENT)
OF STATE et al.,)
)
Defendants.)
)
)
_____)

I, Michael Paul Johnson, hereby declare:
1. I am a computer security consultant, and have extensive experience and training in the field of cryptology.
2. Currently, I am the Vice President of Pion Engineering, Inc.; the Director and President of Rainbow Missions, Inc.; and a Co-moderator of the Internet newsgroup sci.crypt.research.
3. I am a Registered Professional Engineer in Colorado, fluent in several computer languages and computer platforms, and have created several cryptographic methods, including the Diamond and Diamond Lite Encryption Algorithms, and the Sapphire II Stream Cipher and the Ruby Hash Cipher.
4. I have published several documents on the subject of data encryption. One of them is the "Where to Get Pretty Good Privacy (PGP) FAQ," which can be found on the Internet newsgroup alt.security.pgp, as well as in "Protect Your Privacy: A Guide for PGP Users" by William Stallings. In addition, I wrote "Cryptology in Cyberspace," which appeared in the October 1995 issue of Cryptologia.
5. I received my BS in Electrical Engineering and Computer Science from the University of Colorado at Boulder, and my MS in Electrical Engineering from the University of Colorado at Colorado Springs. My master's thesis was related to data compression and encryption. In addition, I attended the Navy Nuclear Power School postgraduate training.
6. I would like to emphasize several points:

Cryptography Has Prominent Civil Applications

7. First, cryptographic technology has predominant civil applications such as protection of privacy and prevention of fraud.
8. I use cryptographic technology to protect confidential business information. For example, I encrypt email that contains confidential business information. I also encrypt sensitive personal communications, such as long distance peer counseling. If I did not, I believe would not be properly protecting the privacy of my clients and other correspondents.
9. I use cryptographic technology to implement cryptography export controls on my website.
10. Cryptography allows one to append a digital signature to an encrypted message. A digital signature is analogous to a signed signature in which the message recipient can verify the authenticity of the message sender. I use digital signature to authenticate most of the shareware and electronic documents that I write and make it possible for people to check for corruption, virus infection, or malicious tampering. For example, I have digitally signed a book titled The Good News According to John, God's Living Word Translation to validate its integrity and source.
11. Besides these direct benefits, I receive the indirect benefit of non-military cryptography via secured banking transactions, electronic payment authorizations, and
privacy protection by people who keep personal data about me.
12. Cryptographic technology is also used in academic and scientific exchange of ideas and information.

No Reasonable Distinction Between Algorithms and Source Code

13. Second, the same mathematical algorithms can be used both for authenticating data and encrypting messages. Consequently, the same computer software that is designed to compute digital signatures, a task which is not restricted by the ITAR, can be used to perform public key encryption, a task which is restricted by the ITAR.
14. In fact, there is no reasonable distinction between "software" and "algorithm." Both are descriptions of how to do some mathematical tasks.
15. Let me use the classic one-time-pad algorithm to illustrate these two points:
16. If I want to encrypt a message that includes an alphabetical character , I can convert it to a numeric value (A), and then add this numeric value (A) to a random number (B) known only to myself and the recipient of this message. The sum is the ciphertext (C) that can be transmitted over an insecure channel without compromising the secrecy of the message (A). The recipient then subtracts the secret value (B) from the ciphertext (C) to recover the message (A). A programmer can easily perform this task in several programming languages. If I use the computer language called C and modulo-2 addition, the algorithm is:
C = A ^ B;
17. The recipient can decrypt the encrypted message, by using the same algorithm, and recover the numeric value (A). A = C ^ B;
18. If I add nine lines of computer code for standard administrative tasks, such as opening files, I have the source code of this program.
19. If I compile the source code by simply pressing a button, I have a working program.
20. From algorithm to source code to working program, each stage above can easily be derived from the one above or below it.
21. Because computer languages are much more precise than either natural languages (like English) or the standard mathematical notations to explain computer algorithms, computer
languages are useful for communicating exact algorithms to other humans.

ITAR Has Chilled My Work And Publications In Cryptography

22. Third, the International Traffic in Arms Regulations (ITAR) has chilled my cryptographic development and publication, in at least two instances.
23. The first instance took place a few years back. As part of my master thesis, I developed the MPJ encryption algorithm. Believing that the algorithm was good enough to
merit a broader audience, my academic advisors recommended me to publish the algorithm, in conjunction with the thesis itself. As part of the process, I wrote a shareware encryption program that implemented the algorithm and distributed it electronically.
24. I sent a courtesy copy to the NSA. The NSA informed me that, although I could freely use and distribute my program in the United States, exporting the shareware would be restricted by ITAR.
25. Because shareware simply can not be recalled, and normal shareware distribution channels have

absolutely no export barriers, I was afraid that I might inadvertently violated some regulations, thus putting my job (which at that time required a security clearance), my finances, and even my freedom in danger. Because I did not want to violate or appear to violate any United States laws, I stopped my cryptographic publication activity.

26. The second instance took place about three years ago, when I started work on a computer cryptography book that would include a cryptographic function library on disk. I wanted to comply with all legal requirements. I tried to find out the legal requirements, including the difference between an exportable and non-exportable encryption program. Starting in 1993, I spent many hours of telephone conversation with both Department of State and National Security Agency (NSA) export control officers, who consistently refused to tell me in advance what the criteria were that they used to decide the difference between exportable and non-exportable cryptographic software.

27. For example, on the afternoon of the 15th July, 1994, Brian Rink of the NSA repeatedly refused to tell me what I needed to do to write an encryption program that was exportable under the much less restrictive Department of Commerce rules, insisting that I must submit a completed program that was ready to sell and that they would only tell me if it was exportable or not, with no reasons given either way. Even though Charlotte Knepper of the NSA had earlier told me that they would evaluate a design on a program if I submitted a "Commodity Jurisdiction Request" (CJ), Brian Rink insisted on seeing a full implementation of the program before telling me anything about its exportability. Brian said "We don't evaluate designs." Later, when asking about key size limitations, I asked "If 40 bits wasn't acceptable, would you tell me what key size would be acceptable?" Brian answered, "No. Probably not."

28. The CJ that Brian and I were discussing was returned without action after the 15 business day time limit given by the Department of State had expired, contrary to verbal promises given to me by both the Department of State and the NSA.

29. When I submitted the second official "Commodity Jurisdiction Request" (number 358-94 for "Quicrypt," a completed software product), the NSA representatives again failed to meet the deadlines that they promised by demanding that I make changes to the product on the 15th business day since they acknowledged receipt of the CJ request. I made the exact changes that they requested, documented them, and sent them to them electronically the same day. They delayed an additional 5 working days before issuing a response.

30. The NSA's conditions of (1) keeping my source code for Quicrypt secret and (2) limiting the effective key length of only 32 bits (weaker than a 40-bit key by a factor of 256) crippled the resulting product to something that has not been nearly as successful as it would have been without these restrictions, since I compete with unrestricted products originating outside of the USA.

31. More than one year passed between the time I first started asking and the time I managed to get a favorable response to a "Commodity Jurisdiction Request." Comparing to a programming effort of about two standard work weeks, NSA imposed an unreasonable administrative overhead on my project. They also prevented me from writing much better software by forcing me to divert resources from analysis and programming to many hours of legal research.

32. While I was waiting for the approval of my shareware, some people outside of the USA wrote some very good shareware encryption products, including Secure Device, HPACK, enhancements to PGP, and Peter Gutman's cryptographic toolkit. It is likely that many customers of these products would be using my program, if the program was not delayed and weakened by the NSA and was available in the market on time.

33. The NSA's policy and delay caused me damages, because ideas published can result in consulting business, esteem, and other tangible and intangible benefits. Delay is denial of these benefits during the time of delay. This is especially important in the cryptology field because "proprietary" encryption for which the author will not share the source code is considered insecure. Serious cryptographers will not take the programmer seriously unless he or she shares source code. In the above two instances, the NSA denied me the right to share my source code in a timely manner.

I declare under penalty of perjury that the foregoing is true and correct and that this Declaration was signed at Longmont, Colorado.

Dated: _____ _____

Michael Paul Johnson